AD-A058 821 DOUGLAS AIRCRAFT CO LONG BEACH CALIF F/G 20/4
A THEORETICAL METHOD FOR CALCULATING THE AERODYNAMIC CHARACTERI--ETC(U)
1977 M L LOPEZ, C SHEN, N F WASSON N00014-71-C-0250

UNCLASSIFIED MDC-J5519-02-VOL-2 NL

1 OF 2
AD
A058 821

THE ENCLOSED REPORT SUPERSEDES THE EARLIER EDITION WHICH, ACCORDING TO OUR RECORDS, IS IN YOUR POSSESSION.

Reproduction in whole or in part
is permitted for any purpose of the
United States Government

Approved for public release;
distribution unlimited

DOUGLAS AIRCRAFT COMPANY

MCDONNELL DOUGLAS CORPORATION

NOMENCLATURE

(12) 128p.

(4)

ISSUED May 73
SECOND EDITION 1977

(14) REPORT NO. MDC-J5519-02-VOL-

(11) 1977

(6) A THEORETICAL METHOD FOR CALCULATING
THE AERODYNAMIC CHARACTERISTICS
OF ARBITRARY JET FLAPPED WINGS Volume II. EVD Jet-Wing Computer Program User's Manual (CDC 6000 Series Computers. Second Edition.

(10) Michael Cheng-Chung Norman
M. L. Lopez, C. C. Shen and N. F. Wasson

VOLUME II (N.H.)

EVD JET-WING COMPUTER PROGRAM
USER'S MANUAL
(CDC 6000 SERIES COMPUTERS)

Prepared by the Douglas Aircraft Company,
Long Beach, California under Contract
(15) N00014-71-C-0250/NR 215-189 (Code 461),
Sponsored by the Office of Naval Research, Aeronautics

(9) Final technical rept.

116 400

This report describes the EVD Jet-Wing Computer Program, developed by the V/STOL Technology Development Group - Aerodynamics, of the Douglas Aircraft Company, McDonnell Douglas Corporation. Development of the program was begun in 1970 under sponsorship of the McDonnell Douglas Independent Research and Development Program (IRAD), and resulted in the Mark I version. From April, 1971 through April, 1972, the program was entirely rewritten, drawing heavily on the basic features developed in the Mark I version, but incorporating extensive improvements in program flexibility, efficiency and the amount of information available to the user. This work, culminating in the Mark II version of the program, was sponsored by Office of Naval Research Contract N00014-71-C-0250.

The major portion of the program was written by Mr. N. F. Wasson under the direction of Dr. C. C. Shen and Mr. M. L. Lopez, Technical Director. Other programming contributions were made by Mr. N. D. Halsey on the momentum "induced drag" method, and by Messrs. J. L. Hess, T. M. Ridell, and D. N. Smyth on the matrix solution. The technical work of Messrs. M. I. Goldhammer, B. K. Lakin, and W. V. Whitman in applying the program to various configurations has also been of great value in its design, verification and correction.

This report has been reviewed and is approved.

M. L. Lopez, Section Manager
V/STOL Systems
Aerodynamic Technology

E. S. Rutowski, Chief
Aerodynamics Engineer
Technology Development

O. R. Dunn, Director
of Aerodynamics

ABSTRACT

This report describes the EVD Jet-Wing Computer Program, which is based upon the Elementary Vortex Distribution (EVD) Jet-Wing Lifting Surface Theory described in Volume I of this report. This program provides a capability for determining the aerodynamic characteristics of wings of arbitrary planform, and includes the following:

1. Spanwise and chordwise loading
2. Spanwise variation of induced drag
3. A capability to investigate the effects of:
   a. Part span flaps
   b. Part span blowing
   c. Pitching, rolling, yawing and sideslip
4. Total lift and induced drag (momentum method), pitching, yawing and rolling moments, etc.

The program has the capabilities for investigating the effects of a variation of leading and trailing flap deflection, camber, twist, jet deflection and jet momentum.

PREFACE TO REVISED EDITION

This report was originally published in May 1973 in two volumes: Volume I, The Elementary Vortex Distribution Jet-Wing Lifting Surface Theory and Volume II, EVD Jet-Wing Computer Program User's Manual. In this new edition several typographical errors have been corrected, a minor change made in the computer program, and the "Limited Distribution" of Volume II removed.

Michael L. Lopez
Douglas Aircraft Company
McDonnell Douglas Corporation

CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

INDEX OF FIGURES

NOMENCLATURE

NAME	SYMBOL	INPUT/ OUTPUT	DEFINITION
A	A	I/Ø	Fundamental case scale factor
AC	ε_c	I/Ø	Local incidence due to camber
ACTE	ε_{ct}	I/Ø	Local incidence due to camber at trailing edge
ALFIN	α_{i_∞}	Ø	Downwash angle or jet angle at infinity downstream
ALPHA	α	Ø	Angle of attack
ARATIØ	AR	I/Ø	Aspect ratio
AREA	S	I/Ø	Wing planform reference area
BETA	β	I/Ø	Deflection angle, (δ_f, δ_J, or δ_s)
CBGL	$C_{B_{\Gamma_L}}$	Ø	Root bending moment coefficient due to pressure on left wing
CBGR	$C_{B_{\Gamma_R}}$	Ø	Root bending moment coefficient due to pressure on right wing
CBJL	$C_{B_{J_L}}$	Ø	Root bending moment coefficient due to jet reaction on left wing
CBJR	$C_{B_{J_R}}$	Ø	Root bending moment coefficient due to jet reaction on right wing
CBL	C_{B_L}	Ø	Total root bending moment coefficient on left wing
CBR	C_{B_R}	Ø	Total root bending moment coefficient on right wing
CCD	$(C_{D_i})_p$	Ø	Total induced drag coefficient calculated by pressure integration
CCJ	C_J	Ø	Total jet momentum coefficient
CCL	C_L	Ø	Total lift coefficient
CCM	C_m	Ø	Total pitching moment coefficient (about wing apex)
CCS	C_S	Ø	Total leading edge suction coefficient

NOMENCLATURE

NAME	SYMBOL	INPUT/ OUTPUT	DEFINITION
CCT	C_T	Ø	Total net thrust coefficient
CCY	C_Y	Ø	Total side force coefficient
CD	c_{d_i}	Ø	Sectional induced drag coefficient
CDITZ	$(C_{D_i})_M$	Ø	Total induced drag coefficient, calculated by the momentum method
CHØRD	c	Ø	Wing sectional chord
CL	c_ℓ	Ø	Sectional lift coefficient
CLL	C_l	Ø	Rolling moment coefficient
CLLP	C_{l_p}	Ø	Rolling moment coefficient derivative due to rolling
CLLR	C_{l_r}	Ø	Rolling moment coefficient derivative due to yawing
CLQ	C_{L_q}	Ø	Lift coefficient derivative due to pitching
CM	c_m	Ø	Sectional pitching moment coefficient (about the local leading edge)
CMAC	$\bar{\bar{c}}$	Ø	Mean aerodynamic chord (MAC)
CMQ	C_{m_q}	Ø	Pitching moment coefficient derivative due to pitching about center of gravity
CMU	c_μ	I/Ø	Sectional jet momentum coefficient
CN	C_n	Ø	Yawing moment coefficient
CNI	C_{n_i}	Ø	Total yawing moment coefficient due to induced drag (including leading edge suction)
CN(P)	C_{n_p}	Ø	Yawing moment coefficient derivative due to rolling, dependent on rolling rate
CN(R)	C_{n_r}	Ø	Yawing moment coefficient derivative due to yawing, dependent on yawing rate
CP	Δc_p	Ø	Pressure coefficient, $\Delta c_p = c_{p_{\text{lower surface}}} - c_{p_{\text{upper surface}}}$

NOMENCLATURE

NAME	SYMBOL	INPUT/ OUTPUT	DEFINITION
CPMBL	$\frac{Y_{C.L.}}{b/2}$	Ø	Spanwise center of lift of left wing, in terms of half span
CPMBR	$\frac{Y_{C.L.}}{b/2}$	Ø	Spanwise center of lift of right wing, in terms of half span
CREF	$\tilde{c}$	I/Ø	Wing reference chord
CS	c_s	Ø	Sectional leading edge suction coefficient
CTO	c_t	Ø	1. Sectional thrust coefficient 2. Tip chord of wing
CXCL	$\frac{X_{C.L.}}{\tilde{c}}$	Ø	x-coordinate of center of lift, in terms of reference chord CXCL = - (CCMG + CCML)/CCL labeled (X/CREF)
CXCLB	$\frac{X_{C.L.}}{b/2}$	Ø	x-coordinate of center of lift, in terms of half span CXCLB = - ((CCMG + CCML)/CCL)* CREF labeled (X/B/2)
CXCP	$\frac{X_{C.P.}}{\tilde{c}}$	Ø	x-coordinate of center of pressure, in terms of reference chord
CXCPB	$\frac{X_{C.P.}}{b/2}$	Ø	x-coordinate of center of pressure, in terms of half span
CY(P)	C_{Y_p}		Side force coefficient derivative due to rolling, dependent on rolling rate
CY(R)	C_{Y_r}	Ø	Side force coefficient derivative due to yawing, dependent on yawing rate
D	d	Ø	Chordwise distance from trailing edge to infinity EVD vortex point
DEL	$\bar{\delta}$	Ø	Chordwise length of an EVD element in terms of local chord
DELTA	$\bar{\Delta}$	Ø	Half of the spanwise width of an EVD element or a spanwise division, normalized by b/2
DJ	δ_J	I/Ø	Jet deflection angle relative to the wing trailing edge slope

NOMENCLATURE

NAME	SYMBOL	INPUT/ OUTPUT	DEFINITION
EPS	ε_i	Ø	Total local incidence angle at a downwash control point on the wing
GAMMA	γ	Ø	Vorticity intensity
HL	h_ℓ	I/Ø	Height of leading edge above the x-y plane
I	i	Ø	EVD element sequence number
ICT		I	Camber type flag for each wing section
ICTYPE		I	Wing chordwise division type flag for each section
IDERIV		I	Stability derivative flag
IGTYPE		I/Ø	Planform geometry type flag
IHINGE		I/Ø	Hinge EVD flag
IHT		I	Hinge type flag for each wing section
IJTYPE		I	Jet chordwise division type flag for each section
ILT		I	Flap type flag
INBETA		I	Hinge angle input flag
INCAMB		I	Camber angle input flag
INDELJ		I	Jet angle input flag
INHITE		I	Leading edge height input flag
INTWST		I	Twist angle input flag
IPRINT		I/Ø	Output control flag
ISYMM		I/Ø	Symmetry control flag
J	J		Sectional jet momentum
JETFLG		I/Ø	Jet control flag
K	k		Sequence number of spanwise divisions or sections

NOMENCLATURE

NAME	SYMBOL	INPUT/ OUTPUT	DEFINITION
N		I/Ø	Fundamental case sequence number
NCASES	K	I/Ø	Number of fundamental cases
NCT		I	Number of camber types
NHT		I	Number of hinge types
NI		I	Number of chordwise divisions of a particular chordwise division type (wing or jet)
NJ		Ø	Number of chordwise divisions of jet section
NJTYPE		I	Number of jet chordwise division types
NRØWS		Ø	Number of wing sections or spanwise divisions
NRØWSJ		I	Number of sections having a trailing jet
NW		Ø	Number of chordwise division of each section
NWTYPE		I	Number of wing chordwise division types
SPAN	b	I/Ø	Wing span
SWEEP	$\Lambda_{c/4}$	I/Ø	Wing quarter-chord sweep angle
TANLE	$\tan\Lambda_{\ell}$	Ø	Tangent of sectional leading edge sweep angle
THETA	θ	Ø	Total jet deflection angle relative to the free stream
TITLE		I/Ø	Run title
TR	λ	I	Wing taper ratio
TWIST	α_o	I/Ø	Twist angle for each wing section
TYPE		Ø	EVD type flag
U	U		Free stream velocity
X	x	Ø	Chordwise coordinate of the Cartesian coordinate system

NOMENCLATURE

NAME	SYMBOL	INPUT/ OUTPUT	DEFINITION
XB	$\bar{x}$	I/Ø	Chordwise distance of an EVD element from the leading edge, in terms of local chord
XBH	$\bar{x}_h$	I/Ø	Chordwise distance of a hinge point from the leading edge, in terms of local chord
XCL/C	$\bar{x}_{c.l.}$	Ø	Chordwise location of center of lift at a section measured from local leading edge, in terms of local chord
XCP/C	$\bar{x}_{c.p.}$	Ø	Chordwise location of center of pressure at a section measured from local leading edge, in terms of local chord
XCG	$x_{c.g.}$	I/Ø	x-coordinate of the center of gravity about which pitching rate derivatives are taken
XI	x_i	Ø	x-coordinate of an EVD element
XLEAD	x_ℓ	I/Ø	x-coordinate of the leading edge
XMC	$x_{m.c.}$	I/Ø	x-coordinate of the moment center about which pitching moments are taken
XTRAIL	x_t	I/Ø	x-coordinate of the trailing edge
Y	y	I/Ø	y-coordinate of a section

Subscripts

SUFFIX	SUBSCRIPT	DEFINITION
A	α	Indicates a linear variation with angle of attack
A2	α^2	Indicates a quadratic variation with angle of attack
G	Γ	Indicates a contribution due to pressure (circulation)
J	J	Indicates a contribution due to jet reaction
MC	m.c.	Indicates pitching moments taken about moment center
MU	μ	Indicates a contribution due to jet reaction at a wing section

NOMENCLATURE

SUFFIX	SUBSCRIPT	DEFINITION
R2	r^2	Indicates a quadratic variation with yawing rate
T	t	Indicates a contribution due to the thrust component of the jet reaction at a section
X	α	Indicates a contribution from mutual interference between the basic configuration and the angle of attack
0	o	Indicates $\alpha = 0$

1.0 INTRODUCTION

This report describes the Mark II version of the EVD Jet-Wing Computer Program, intended for use as a preliminary design tool for the analysis of arbitrary wings incorporating trailing edge blowing. The program is based upon the linearized Elementary Vortex Distribution (EVD) Jet-Wing Lifting Surface Theory described in Volume I of this report (Reference 1). The program is written in the Fortran IV language and can be adapted for use on many existing large computing sytems. Versions are currently in operation on the CDC 6000 series and IBM 360 and 370 series computers.

In order to execute the program, the user must describe the arbitrary geometric configuration to the computer by using several different types of input cards containing specific information in a fixed order. Deflection, for example, of a wing control surface may be accomplished by using the linear "fundamental case" feature of the program. Superposition of the aerodynamic characteristics associated with fundamental cases so as to provide the overall aerodynamic characteristics of a complete configuration may be obtained by specifying the "composite case" feature of the program. The number of jet strength cases which can be produced for each geometric configuration is limited only by the computing time available. Printed output from the program contains the jet-wing aerodynamic characteristics in coefficient form. Detailed spanwise and chordwise loading and certain dynamic stability derivatives may also be included at the option of the user.

Maximum realization of the potential of this program can only be achieved by a user who is thoroughly familiar with the applications and limitations of the linearized theory upon which it is based, and who is willing to apply his own engineering knowledge and judgment in a methodical manner to each configuration analyzed. It is suggested that both volumes of this report be read before running the first problem, and that a few simple problems be tried in order to develop understanding and familiarity with the program before large complicated problems are attempted.

2.0 FUNDAMENTALS OF PROGRAM USE

The EVD jet-wing computer program is based upon the linearized EVD jet-wing lifting surface theory described in Reference 1. This program has several unique features which enable it to be used by the aerodynamicist in the evaluation, design and development of V/STOL aircraft systems which utilize powered high lift systems based on the jet flap principle (e.g., EBF, augmentor wing, etc.). The scope of this program is not limited, however, to the analysis of jet flap wings. By selecting several options available to the user, this program may be used as a classical lifting surface method. Hence, as an aerodynamic tool, the analysis of conventional wings can be regarded as being within the scope of the present computer program.

The program is written entirely in the Fortran IV language, and can be adopted for use on any large-scale computer system. The program consists of several primary components designed to minimize computer storage requirements while maintaining maximum computer efficiency and speed. This modular arrangement also results in easier program checkout and correction.

The programming philosophy has sought to maintain the independence of each small component of the program wherever possible. This approach also facilitates the addition of new capabilities, with minimum alteration of the existing components.

2.1 Philosophy of Program Use

It is fundamentally important that the user of the EVD jet-wing computer program understand the basic limitations and approximations adopted in the solution of the jet-wing problem. The program is not a "magic box" with which he can expect to achieve valid results while supplying poorly prepared input data. He must, therefore, understand each step of his role in the analysis of a specific problem, including use of engineering knowledge and judgement in every decision he makes in the preparation of computer input data. Only then can he expect the program, which is essentially an engineering tool, to reliably and accurately complete the task assigned to it. This point may be summarize by the quaint term which applies to all computer programs: "GIGO" - Garbage In, Garbage Out!

Judgement is also required in the interpretation and application of the computer output data. The user should be aware of those factors not included in the program analysis, which may have an effect on the final result. He should also have some understanding of the way in which these factors would be likely to influence the program results. For example, if there are regions of flow separation on a real wing, the program would be expected to overpredict the lift of that wing. On the other hand, a thick wing is expected to have higher lift than the wing of zero thickness which is considered by the present program. The quantitative effects of such factors are, of course, very difficult to accurately predict, but the user should at least be aware of the trends which they are expected to produce.

The user is also cautioned against extrapolating the linearized results into regions where strong nonlinear behavior might be expected. For example, the small angle assumptions of the linearized approach make it unlikely that the program would accurately predict the characteristics of a wing with, say, a flap deflected at 60 degrees. While the program would give an indication of the trends produced by such a nonlinear configuration, and in certain cases linearized results have been shown to be remarkably good in seemingly nonlinear situations, it would nevertheless be dangerous to use the output data as though it were generally an accurate prediction of the aerodynamic characteristics under all conditions.

2.2 Brief Review of the Linearized Approach

The assumptions and restrictions of the EVD jet-wing lifting surface theory are discussed in detail in Volume I of this report. For the convenience of the reader, the basic assumptions are restated here:

On the Wing:

a. The wing is thin, and is represented by the mean camber line.

b. All local incidences are small, but may be discontinuous.

On the Jet:

a. The jet is thin.

b. The jet deflection, relative to the freestream, is small.

General Flowfield:

a. Incompressible, inviscid, irrotational.

b. No mixing occurs between the jet and the external flow.

c. All spanwise components of velocity are considered small relative to the freestream velocity.
d. Rollup of the jet sheet and wing and jet trailing vortex system is neglected.

The above assumptions imply the existence of certain other conditions which are of practical importance in the use of the program. Though they may not seem to be obviously derived from the basic assumptions, these conditions nevertheless are controlling factors in the application of the method, and must be recognized and considered by the user. Such conditions include the following:

a. The wing and jet are represented by planar sheets of bound trailing vorticity.
b. All potential field influences of the wing and jet originate in the wing and jet plane.
c. All boundary conditions are satisfied in the wing and jet plane.
d. The jet issues from the trailing edge of the wing, and its reaction force acts there also.
e. Since the lifting surface theory is for a jet-wing system alone, no account is taken for the effects of fuselage, nacelles, empennage, etc.
f. There are no losses or dissipation associated with the jet sheet emergence from the wing trailing edge.
g. There are no gaps or slots in the wing (program restriction only).

2.3 Geometry Preparation

The EVD jet-wing computer program is capable of evaluating the aerodynamic characteristics of jet-wings of arbitrary wing planform and arbitrary jet spanwise distribution of momentum. In all cases, the jet sheet correctly extends to infinity downstream. The jet-wing system may be considered as symmetric, anti-symmetric, or non-symmetric. A symmetric jet-wing is defined as one for which, under all conditions, the computed aerodynamic loading would

be symmetric about the x-axis. This implies, therefore, symmetry about the x-axis of all the following characteristics:

Wing:

a. Planform geometry

b. Local incidence of each element

Jet:

c. Spanwise location

d. Strength at each spanwise section

e. Deflection

Anti-symmetric jet-wings are defined as having anti-symmetric computed aerodynamic loading. Thus, they must be symmetric in all of the above items except for (b) and (e), which must be anti-symmetric. Since anti-symmetry in the angle of attack of the wing is meaningless, the program will always treat the anti-symmetric jet-wing as being at zero angle of attack only. If an otherwise anti-symmetric wing were at an angle of attack, its loading would be non-symmetric, and it would, therefore, have to be treated as completely non-symmetric. Non-symmetric jet-wings are defined as having non-symmetry in any one or more of the characteristics listed in Items (a) through (e) above.

It is, however, possible for the user to superimpose by hand the results of anti-symmetric runs with those of symmetric or non-symmetric runs. Such a situation may arise, for example, when a quantity of data has been generated for various configurations of a particular wing and the effects of anti-symmetric ailerons are then required. It may be more efficient to compute the aileron effects alone in one additional run rather than rerunning all the previous configurations non-symmetrically with ailerons included. The superimposed loading will be in general non-symmetric and both the right and left halves of the wing must be considered. The linear aerodynamic coefficients (e.g., C_L, C_M, C_l) may be superimposed directly at zero angle of attack, and the variation with angle of attack will be the original symmetric or non-symmetric values. The non-linear coefficients (e.g., C_D, C_Y, C_n) cannot be superimposed directly. Instead, they must be obtained by an integration of the chordwise and spanwise loading.

The problem analysis begins with the user definition of the planform geometry of the wing and jet. The wing and jet are broken up into spanwise sections of arbitrary width. The arrangement of these sections and their size are mainly determined by two factors. First is the occurrence of planform discontinuities; for example, flap edges, flap extensions, leading and trailing edge brakes, jet limits, etc. Second is the allowance for spanwise variation of loading. Since the program loading is assumed to be that which occurs at the center of each section and is constant in the spanwise direction on each section, there should be close spanwise spacing in regions where rapid loading variation might be expected. For example, near the wing tip or near the edge of a highly loaded flap or strong jet, there will normally be a rapid change in spanwise loading. The spanwise sections should, therefore, be narrow in these regions so that the "stairstep" loading of the program can adequately represent a smooth loading variation.

Each spanwise section is further broken up into chordwise divisions so that the jet-wing planform is represented by an array of rectangular elements with which are associated an equal number of Elementary Vortex Distribution elements. This representation is shown in Figure 1 for a typical jet-wing. Note that no two adjacent elements overlap, nor are there any gaps. Also, that any one of four types of EVD elements, (i.e., Regular, Leading Edge, Hinge, and Infinity), can be selected.

Given close uniform spacing, the program would always yield a good, smooth chordwise loading. But the linear nature of loading given by the Regular EVD elements allows much larger spacing to be used in regions where the loading is expected to nearly linear, that is, where the slope of the loading is nearly constant. In addition, in order to more accurately approximate the chordwise loading, special EVD elements have been developed. These EVD elements provide an accurate representation of chordwise loading in these special regions which are normally difficult to approximate with the Regular EVD. These regions are near the leading edge, near the abrupt turning angle produced at a flap or jet "hinge" point, and far downstream on the jet sheet. Use of the special EVD elements makes chordwise spacing less critical and reduces the required number of elements on each section. Most of the chordwise EVD elmenents are of the regular type, representing a linear loading

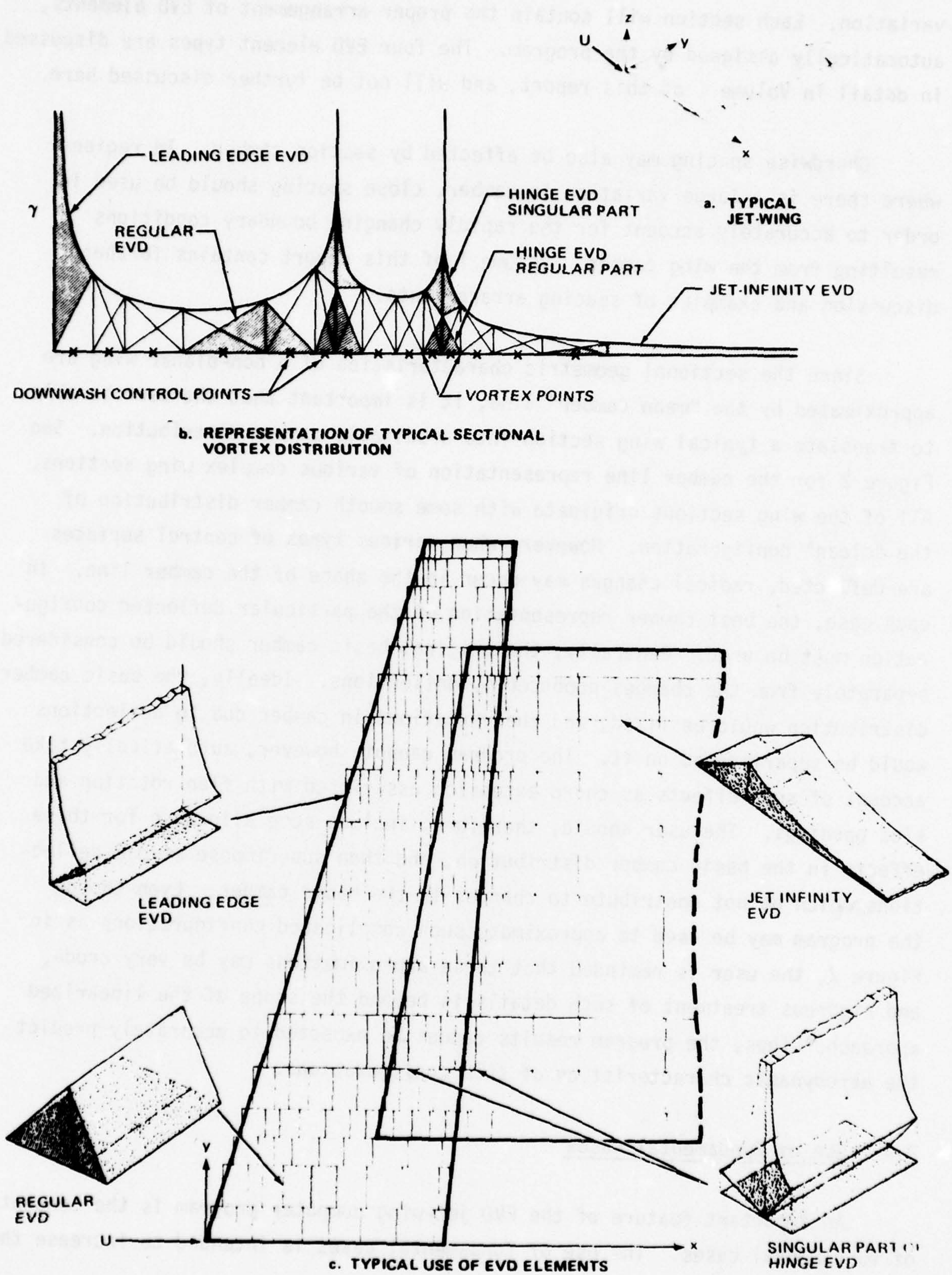


FIGURE 1. PROGRAM REPRESENTATION OF A JET WING

variation. Each section will contain the proper arrangement of EVD elements, automatically assigned by the program. The four EVD element types are discussed in detail in Volume I of this report, and will not be further discussed here.

Chordwise spacing may also be affected by section camber. In regions where there is a large variation in camber, close spacing should be used in order to accurately account for the rapidly changing boundary conditions resulting from the wing camber. Volume I of this report contains further discussion and examples of spacing arrangements.

Since the sectional geometric characteristics of a non-planar wing are approximated by the "mean camber" line, it is important that the user be able to translate a typical wing section into a suitable camber distribution. See Figure 2 for the camber line representation of various complex wing sections. All of the wing sections originate with some smooth camber distribution of the "clean" configuration. However, when various types of control surfaces are deflected, radical changes may occur in the shape of the camber line. In each case, the best camber representation of the particular deflected configuration must be used. Generally, the original basic camber should be considered separately from the changes produced by deflections. Ideally, the basic camber distribution would be fixed, and the variations in camber due to deflections would be superimposed on it. The program cannot, however, automatically take account of such effects as chord extension associated with flap rotation and slot openings. The user should, therefore, include some allowance for those effects in the basic camber distribution, and then superimpose simple deflections which do not contribute to changes in the basic camber. Even though the program may be used to approximate such complicated configurations as in Figure 2, the user is reminded that these approximations may be very crude, and rigorous treatment of such details is beyond the scope of the linearized approach. Thus, the program results cannot be expected to accurately predict the aerodynamic characteristics of such configurations.

2.4 Use of Fundamental Cases

An important feature of the EVD jet-wing computer program is the concept of fundamental cases. The use of fundamental cases is intended to increase the

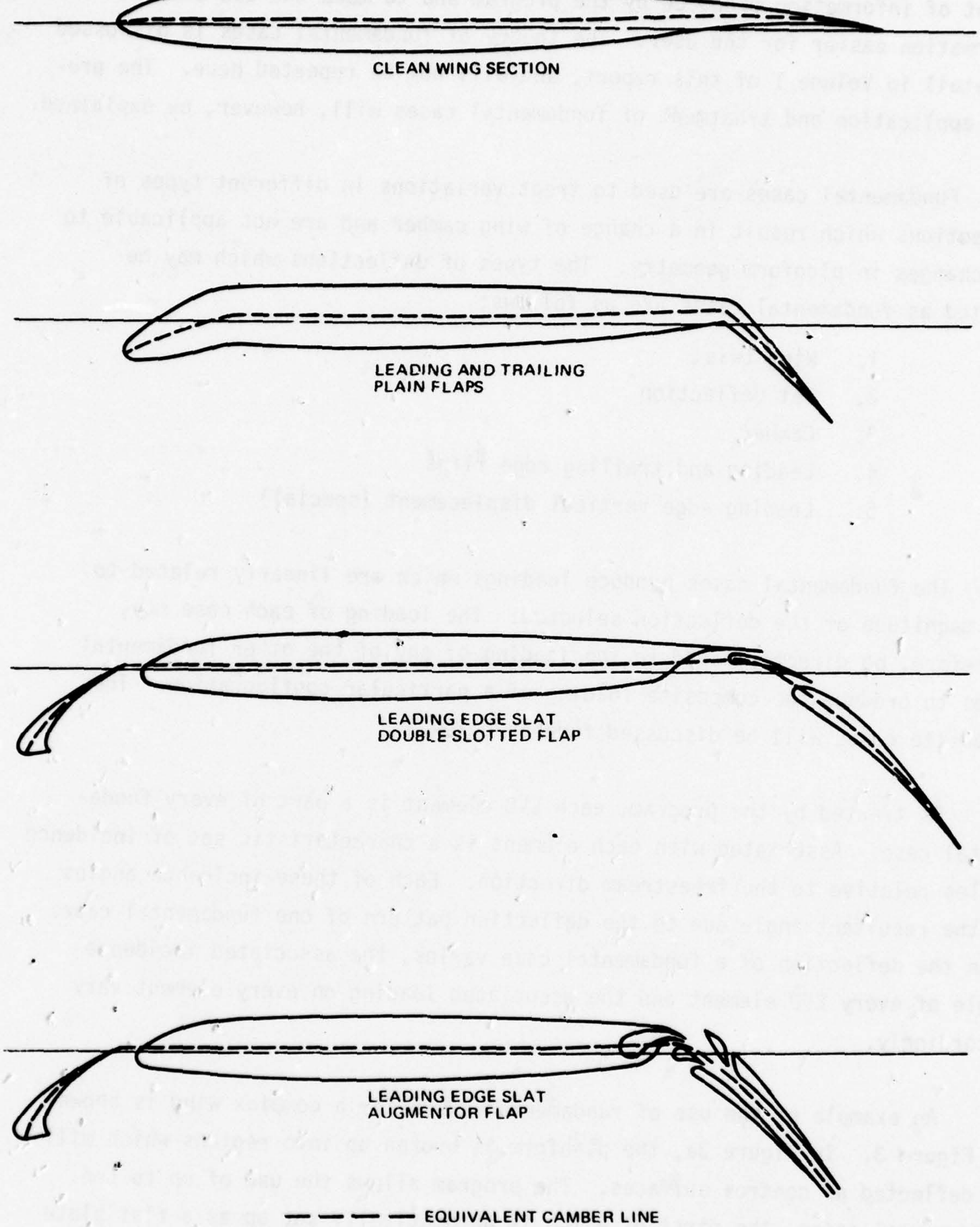


FIGURE 2. REPRESENTATION OF WING SECTIONS BY EQUIVALENT CAMBER

amount of information produced by the program and to make the use of that information easier for the user. The theory of fundamental cases is discussed in detail in Volume I of this report, and will not be repeated here. The program application and treatment of fundamental cases will, however, be explained.

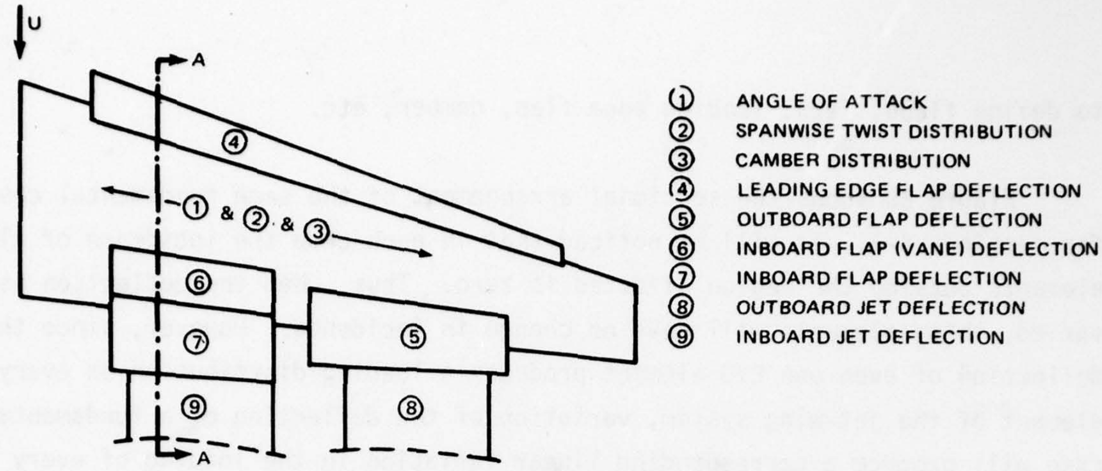
Fundamental cases are used to treat variations in different types of deflections which result in a change of wing camber and are not applicable to any changes in planform geometry. The types of deflections which may be treated as fundamental cases are as follows:

1. Wing twist
2. Jet deflection
3. Camber
4. Leading and trailing edge flaps
5. Leading edge vertical displacement (special)

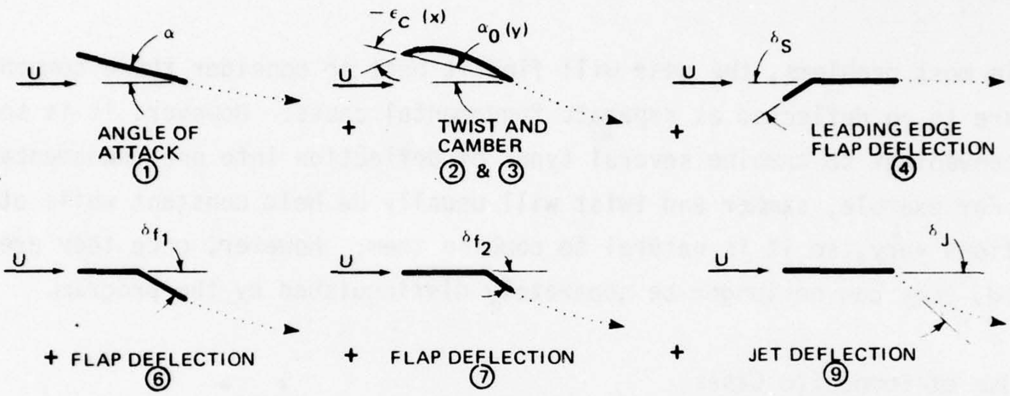
The fundamental cases produce loadings which are linearly related to the magnitude of the deflection selected. The loading of each case may, therefore, be directly added to the loading of any of the other fundamental cases to produce the composite loading of a particular configuration. The composite cases will be discussed further below.

As treated by the program, each EVD element is a part of every fundamental case. Associated with each element is a characteristic set of incidence angles relative to the freestream direction. Each of these incidence angles is the resultant angle due to the deflection pattern of one fundamental case. When the deflection of a fundamental case varies, the associated incidence angle of every EVD element and the associated loading on every element vary accordingly.

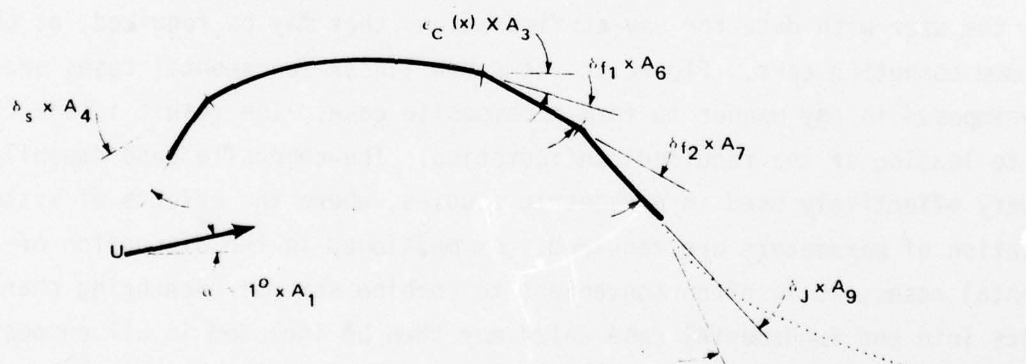
An example of the use of fundamental cases for a complex wing is shown in Figure 3. In Figure 3a, the planform is broken up into regions which will be deflected as control surfaces. The program allows the use of up to ten fundamental cases, the first of which is automatically set up as a flat plate wing at 1° angle of attack. The remaining nine fundamental cases may be used in any manner, and eight of them have arbitrarily been used in the figure



a. ARRANGEMENT OF FUNDAMENTAL CASES FOR A COMPLEX JET WING



b. SEPARATE FUNDAMENTAL CASES OF A TYPICAL WING SECTION (SECTION A-A)



$$\text{COMPOSITE CASE DEFLECTION} = \sum_{i=1}^{\text{NCASES}} \text{FUNDAMENTAL CASE FACTOR } A_i \times \left(\text{FUNDAMENTAL CASE DEFLECTION}\right)_i$$

$$\text{COMPOSITE CASE LOADING} = \sum_{i=1}^{\text{NCASES}} \text{FUNDAMENTAL CASE FACTOR } A_i \times \left(\text{FUNDAMENTAL CASE LOADING}\right)_i$$

c. FORMATION OF A TYPICAL COMPOSITE CASE (SECTION A-A)

FIGURE 3. TREATMENT OF FUNDAMENTAL AND COMPOSITE CASES

to define flaps, jets, leading edge flap, camber, etc.

Figure 3b shows the sectional arrangement of the same fundamental cases for section A-A. It will be noticed that in each case the incidence of all elements outside the region affected is zero. Thus when the deflection is varied, these elements will have no change in incidence. However, since the deflection of even one EVD element produces a loading distribution on every element of the jet-wing system, variation of the deflection of a fundamental case will produce a corresponding linear variation in the loading of every EVD element.

In most problems, the user will find it best to consider those components which are to be deflected as separate fundamental cases. However, it is sometimes convenient to combine several types of deflection into one fundamental case. For example, camber and twist will usually be held constant while other deflections vary, so it is natural to combine them. However, once they are combined, they can no longer be separately distinguished by the program.

2.5 Use of Composite Cases

The purpose of the composite case capability in the program is to provide the user with data for any configurations that may be required, at the minimum computing cost. Figure 3c shows how linear fundamental cases are superimposed in any manner to form a composite case. The result is the composite loading of the required configuration. The composite case capability is very effectively used in parametric studies, where the effects of systematic variation of parameters are required. As mentioned in the discussion of fundamental cases, it is often convenient to combine several unchanging characteristics into one fundamental case which may then be included in all composite cases as a "basic wing" starting point. The user supplies the scale factors by which the program is to multiply the incidence and loading of each respective fundamental case before adding them to get the composite incidence and loading. The user is again cautioned, when using composite cases, about exceeding the range of deflections where linear behavior is likely. He should also be careful, when approximating complex sectional geometric characteristics

such as those of a slotted flap system, to limit the range of deflections so as not to significantly change the planform by chord extension.

2.6 Final Comments

The accuracy of the results obtained with this computer program is generally increased as the number of EVD elements increases, assuming that the elements are wisely located by the user. When pursuing accuracy, however, the relative significance of factors which are unaccounted for must be kept in mind. In addition, the computing time is very strongly affected by the number of elements (somewhere between a quadratic and cubic function of the number of elements). Thus, a compromise must be reached between the desire for the best program results on the one hand and, on the other, the practical limitations of computing resources and various unpredictable influences.

It is again strongly recommended that the user read this computing manual thoroughly before attempting the first problem, and also that he read Volume I for valuable background material and insight into the theoretical foundations on which the program is based.

3.0 PROGRAM DESCRIPTION

3.1 General Description

The McDonnell Douglas EVD Jet-Wing Lifting Surface Program is composed of several basic component programs, each of which is independent of the others in logical program flow. These components rely, however, on each other for sharing of information, and all of them are required for satisfactory analysis of a problem. The component programs are tied together by two master routines, only one of which is used in any given run. The choice of which master routine to execute is made by the main routine according to the user's requirements. Figure 4 shows the general functional arrangement of the various program components. The first master routine, called APPLY1, controls execution of ordinary runs, without stability derivative calculations. If stability derivatives are requested by the user, the main routine calls APPLY2, which controls execution for stability derivative runs. Each of the master routines calls subroutines in the four component programs as they are needed.

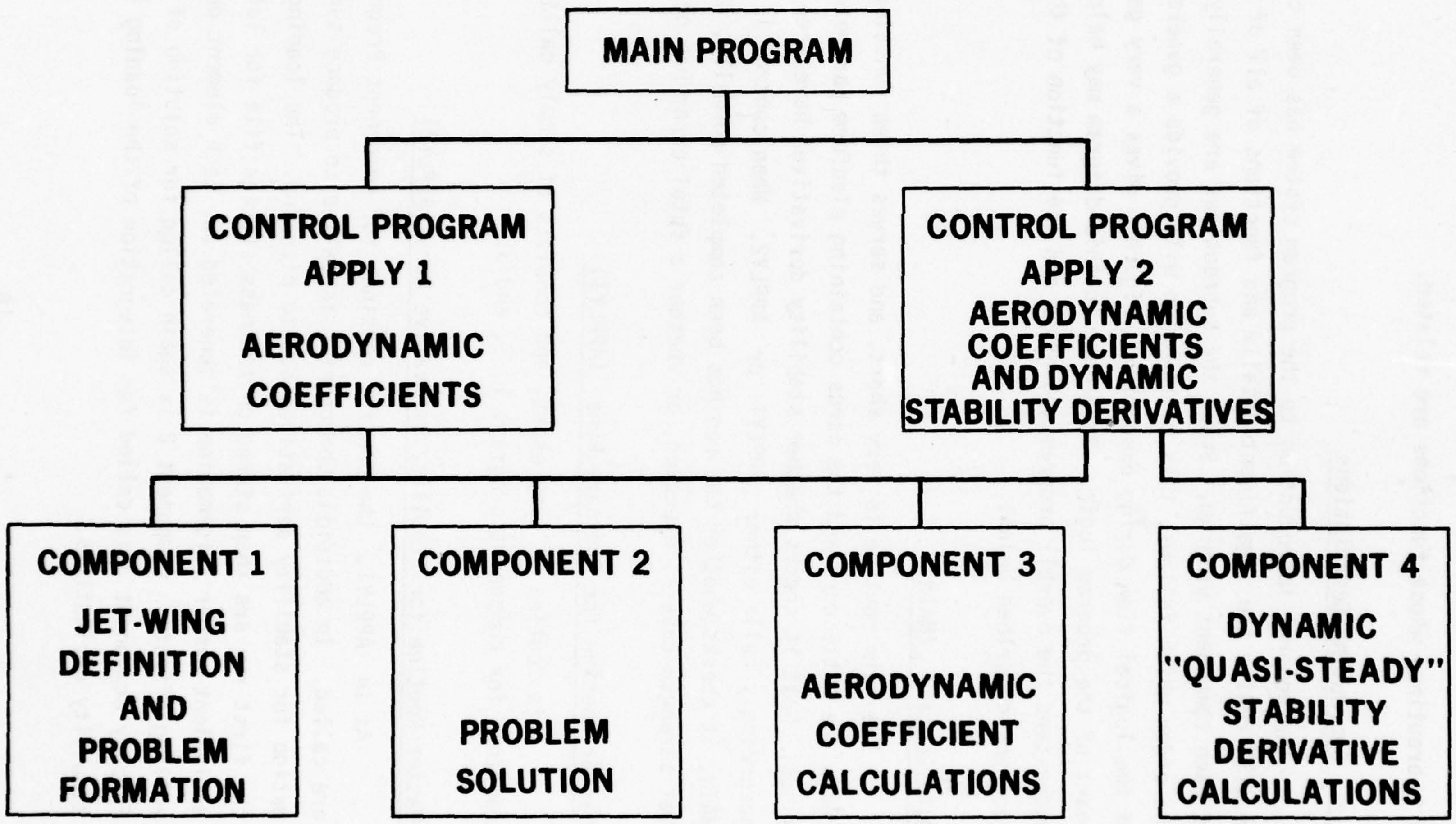
In Component 1, the problem is formed according to the user's input. All input is read, consistency and correctness of input are checked as far as possible, and all geometric parameters are defined.

In Component 2, the mathematical problem is solved. The system of simultaneous linear equations is formed using the downwash influence coefficients of each EVD element. The matrix system is then solved directly, and the solution is achieved in the form of the vorticity strength of each EVD element of the jet wing system.

In Component 3, the vorticity solution is translated into useful aerodynamic coefficients and/or stability derivatives. These include chordwise loading, spanwise distribution of lift, drag and pitching moment, and total lift, drag, and static and dynamic moments.

Component 4 contains supplemental subroutines which are used during calculation of dynamic stability derivatives. This is not a self-contained component program as are the others, but is simply an unstructured collection

FIGURE 4.
GENERAL PROGRAM FUNCTIONAL ARRANGEMENT



of subroutines whose functions are related.

3.2 Subroutine Descriptions

The general introduction to the program system has been completed. Next we will review the significant details and functions of all of the subroutines of each component program. Since the subroutines are generally executed in the order which follows, this review also will provide a general description of the logical flow during execution. Figure 5 gives a very general flow chart of the program logic. Reference to this diagram may help the reader to understand the overall program operation and the function of the various subprograms described below.

Main Routine (MAIN)

The main routine is very short, and serves three functions. It first reads the title card and the cards containing planform parameters and control flags. Next it checks whether stability derivatives have been requested and accordingly calls either APPLY1 or APPLY2. When control is returned to MAIN, it checks whether the run has been completed normally, whether a new jet strength case is expected, or whether a fatal error has occurred.

Master Routine for Ordinary Runs (APPLY1)

This routine is very short, and consists of simply calling the control routines for component programs 1, 2, and 3.

Master Routine for Stability Derivative Runs (APPLY2)

As in APPLY1, the control routines for component Programs 1, 2, and 3 are called. In addition, Component 4 is called to produce supplementary information for stability derivatives due to pitching. The loading solutions for the first run are then stored on the mass storage file for later use, and new equivalent camber information is generated for each element due to yawing and rolling effects. Component 2 is again called for solution of the matrix system. Finally Component 3 is called for integration of the loading to get dynamic stability derivatives.

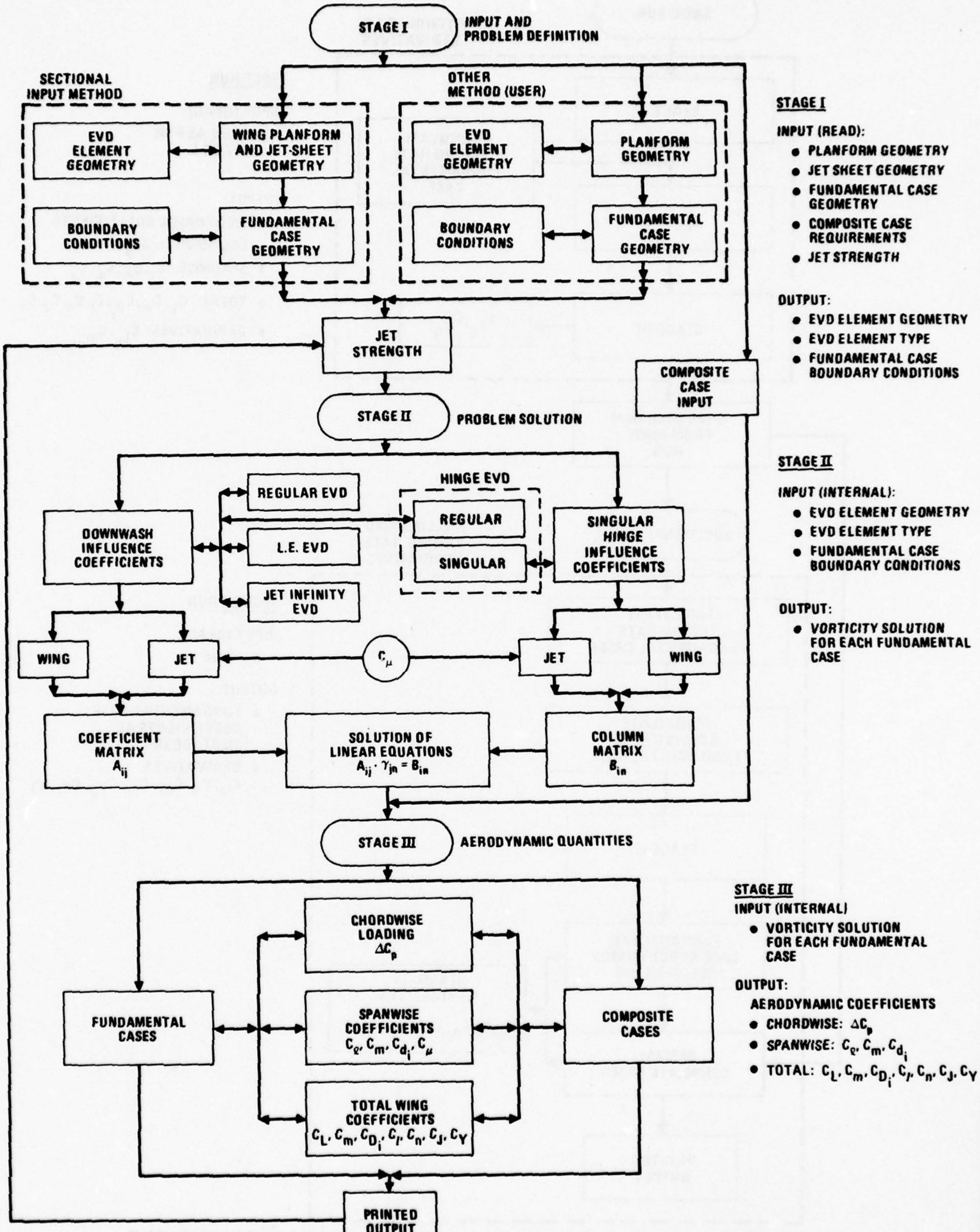


FIGURE 5a. PROGRAM STRUCTURE FOR A BASIC RUN

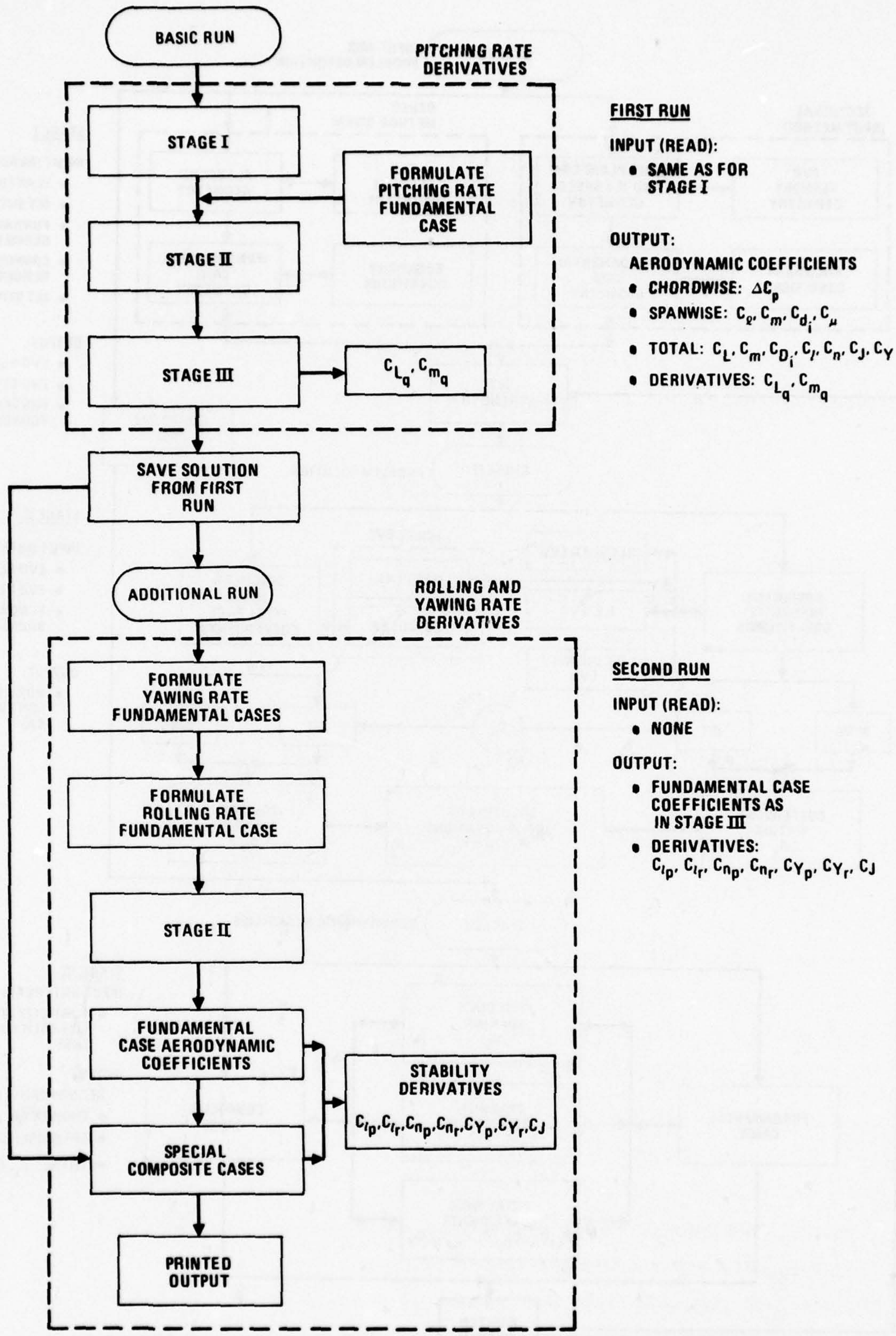


FIGURE 5b. PROGRAM STRUCTURE FOR AN EXTENDED RUN FOR DYNAMIC STABILITY DERIVATIVES

Control of Problem Formation (STAGE1)

The control subroutine (SGMAIN) for dealing with sectional geometry input and element definition is called after the validity of certain control flags has been checked. Alternate methods of input and element definition may be the user-supplied optional routines for any specified values of the IGTYPE flag. (See Section 4.1 - Input Data Instructions). Next the composite case requirements are read by calling INCOMP. Subroutine BLOWIN is called to read the jet strength requirements, and finally BOXJ is called to prepare additional jet parameters for internal use later in the program.

Sectional Input Method Control (SGMAIN)

The spanwise and chordwise spacing of wing elements are read by calling subroutine INPTS. Next the planform is defined by calling either XLETR1 for irregular planforms, or XLETR2 for simple trapezoidal planforms. The planform parameters are normalized by half the input value of wing span in subroutine NORM1. Inputs defining the spanwise and chordwise spacing of elements on the jet are read by calling subroutine INPUTJ. Subroutine BOXS next combines all of the element and planform input data acquired up to this point in order to define the network of elements required by the program. At this point, the jet-wing planform arrangement of EVD elements is completely defined, but it remains to specify the pattern of deflections required by the user. This is accomplished by a sequence of calls to subroutines INCASE, BEECEE, and OUT1 for each required fundamental case. The inputs describing the elements to be deflected and the amount of deflection are read in INCASE. BEECEE interprets this data and derives certain parameters to be used later in the program. OUT1 prints the resulting parameters for each element.

If shortcut methods can be devised for easier input of all or part of the geometrical data required for a class of wings of particular interest, some of the routines above may be replaced, modified or made optional, as the user requires. However, the result must be the same data as produced by the present system in the correct form for use by remaining components of the program.

Wing Element Input (INPTS)

This subroutine reads the spanwise spacing arrangement, and then the number and chordwise spacing of elements on the wing.

Irregular Wing Planform Input (XLETR1)

This subroutine reads the leading and trailing edge input coordinates to define the arbitrary planform, and linearly interpolates where necessary to define the leading and trailing edges at any sections for which no input is given.

Trapezoidal Wing Planform Input (XLETR2)

This subroutine reads the general planform parameters for simple symmetrical trapezoidal wings, and calculates the leading and trailing edge coordinates at each required spanwise section.

Wing Parameter Normalization (NORM1)

This very short subroutine simply normalizes all the dimensional planform parameters by the wing half-span. From this point on, the program deals entirely with the scaled wing, where the wing span is two units.

Jet Element Input (INPUTJ)

In a manner similar to that used in INPTS, the number and chordwise spacing of elements on the jet are read. Note that the spanwise spacing of jet elements is the same as for the wing, but that some or all of these jet sections may have no chordwise elements, thus effectively eliminating the jet at those sections.

EVD Element Definition (BOXS)

This subroutine uses the input data from the above subroutines to compute the length, width, and location of each EVD element on the wing and jet. In addition, the appropriate EVD type is chosen for each element, and several other parameters are defined for later internal use by the program.

Leading and Trailing Edge Sweep (TANS)

This subroutine computes the sweep angle of the leading and trailing edges at each section. This data is needed later for calculation of side force.

Fundamental Case Input (INCASE)

This subroutine reads, for each fundamental case, the flags indicating the types of deflections to be used, then the magnitude of each particular type of deflection. The types of fundamental case deflections read are twist angles, leading edge vertical displacement (for use later in pitching moment calculations), jet deflection angles, camber angles, and hinge location and deflection angles.

Fundamental Case Boundary Conditions (BEECEE)

From the fundamental case input, this subroutine computes, for each fundamental case, the resulting incidence angle of each EVD element relative to the freestream. This is done by summing the accumulative deflections due to twist, camber, and flap deflection from the leading to the trailing edges. The effect of jet deflection is also included on the first jet element of each section. Thus the effects of all deflections are superimposed and the individual effect of, for example, deflection of a particular flap is no longer distinguishable by the program. Therefore, if the user wishes to be able to see the effects of that flap deflection separately from those of the other deflections, he must make use of a separate fundamental case, where only the flap deflection is present.

Fundamental Case Output (OUT1)

This subroutine prints out, for each fundamental case, the location, size, resultant incidence angle, hinge deflection angle, if any, and EVD type of each EVD element. Parameters of spanwise significance are also printed, including wing chord, leading and trailing edge coordinates, leading edge height, twist angle, etc.

Composite Case Input (INCOMP)

This subroutine reads the user's requirements for composite cases, in which the linear fundamental cases may be superimposed in any combination to form a deflection pattern of particular interest.

Jet Strength Input (BLOWIN)

This subroutine simply reads the sectional value of jet momentum coefficient, c_μ, for each section which has been specified to have jet elements.

Additional Jet Parameter Definition (BOXJ)

This subroutine defines an additional jet strength parameter for later internal use, and checks on the consistency of the input c_μ data. The jet strength from the previous c_μ case, if any, is also saved for later internal use.

Control of Problem Solution (STAGE2)

This routine simply calls the two overlay segments which set up the system of simultaneous linear equations and solve the matrix system.

Development of Simultaneous Equations (STG2D)

The large square left-side matrix of downwash influence coefficients is computed by calling DWNWSH. Next this matrix is augmented in SHUFL1. If previous c_μ cases have been run, these steps are skipped and a simple reaugmentation is performed in SHUFL2. For each fundamental case, the right side column of constant boundary conditions is formed in COLUM1. Since the downwash influences of the logarithmic singular hinge EVD elements depend only on the hinge turning angles and thus are known, these influences are computed in HINGE and superimposed directly on the right side matrix in COLUM2. Subroutine PREP then prepares the complete linear equation system for solution by assembling the left and right side matrix data together on tape in the proper form for use by the solution subroutine.

Downwash Influence Coefficients (DWNWSH)

This subroutine calculates the downwash on every element due to the influence of all the other elements. Each element is selected in turn, and the influence of all elements on it are computed one at a time by calling the appropriate downwash influence function: EVD1, EVD2, or EVD3 (hinge EVD not yet considered). If the wing is symmetric or anti-symmetric, the appropriate influence coefficients are superimposed (added or subtracted) to reduce the size of the linear system to half the number of equations. The array of influence coefficients on the particular element under consideration forms one row of the coefficient matrix, and these rows are stored, one by one as they are computed, on the mass storage device, Unit 1. Rows corresponding to elements on the wing are stored first, then rows corresponding to elements on the jet.

Regular Downwash (EVD1)

This function computes the downwash at any point in the wing plane due to a triangular distribution of vorticity on an element located anywhere on the wing (see Figure 1).

Leading Edge Downwash (EVD2)

This function computes the downwash at any point in the wing plane due to a square root singular distribution of vorticity on an element located at the leading edge of the wing (see Figure 1).

Jet Infinity Downwash (EVD3)

This function computes the downwash at any point in the wing plane due to a quadratic decaying distribution of vorticity on a trailing jet element. This element begins several chords behind the wing trailing edge and extends downstream to infinity, where the vorticity decays to zero (see Figure 1).

Hinge EVD (EVD4)

This function computes the downwash at any point in the wing plane due to a logarithmic distribution of vorticity on an element located on the hinge line of a deflected flap, jet, or leading edge flap (see Figure 1).

Initial Matrix Augmentation (SHUFL1)

This subroutine reads the matrix rows corresponding to elements on the jet, subtracts adjacent rows from each other, and further modifies certain elements near the main diagonal due to the nonlinear influence of jet strength. The augmented rows are then replaced on mass storage Unit 1 immediately behind the original downwash matrix rows.

Additional Matrix Augmentation (SHUFL2)

This subroutine reads the augmented matrix rows from Unit 1, remodifies certain elements near the main diagonal, then restores the rows in their original locations.

Column Matrix Formation (COLUM1)

This subroutine defines the right side column matrix of constant boundary conditions. For elements on the wing, the values are the total incidence angles relative to the freestream, in radians. For the first element of each jet section, the value is the total trailing edge incidence angle plus the jet turning angle, in radians, and for all other jet elements, the values are zero. Each column corresponds to one fundamental case.

Hinge Downwash Influence (HINGE)

For each element on the wing and jet, the downwash influence of each hinge singularity distribution is computed by calling EVD4, added or subtracted according to symmetry requirements, then multiplied by the appropriate hinge turning angle. The influence due to all such hinges are summed as they are calculated, so that one hinge influence factor is produced for each element

on the wing and jet. This is done for one column (fundamental case) at a time.

Column Matrix Hinge Augmentation (COLUM2)

This subroutine adds to the right side column matrix row corresponding to each element on the wing and jet, the hinge downwash influence factor computed in HINGE. For rows corresponding to elements on the jet, additional modifications are made to the hinge factors before adding them to the right side matrix in order to account for the influence of jet strength on the vorticity of each jet hinge element. These operations are done for one column (fundamental case) at a time.

Matrix Preparation for Solution (PREP)

The form in which the matrix solution subroutine expects to find data is row by row, left side then right side on each row, stored on a sequential scratch file. Thus the augmented left side rows are read one by one from the mass storage Unit 1 into a "transfer" array, the elements of the right side matrix are defined as the last values of the transfer array, and the entire array is written as one record onto Scratch Unit 2. In this way all the rows corresponding to elements on the wing and then the jet are assembled or "concatenated" and written to Unit 2 to form the entire matrix system.

Control of Matrix Solution (STG2S)

This control routine calls the matrix solution routine, MATRIX, retrieves the vorticity solution, and calls the back substitution check subroutine if requested.

Matrix Solution (MATRIX)

This subroutine solves the matrix system directly by triangularization. It is able to solve very large matrix systems using a relatively small amount of core storage because it deals with only a portion of the system at one time,

the remainder being stored on scratch files.

Scratch File Writing (SAVE)

Because of the large amount of writing to scratch files required in the matrix solution this subroutine is used to write whole arrays as records without reference to subscripts, thus significantly increasing the solution speed.

Scratch File Reading (GETT)

As with subroutine SAVE, reading whole arrays without reference to subscripts significantly reduces the matrix solution time.

Matrix Back Substitution (BAKSUB)

As a check of the matrix solution accuracy, the user may wish to call for a back substitution check. In a row by row manner, the augmented left side matrix is read from mass storage Unit 1 and the corresponding elements are multiplied and summed to form the right side matrix. This right side matrix is printed for reference and should agree with the boundary conditions printed earlier.

Control of Aerodynamic Coefficients (STAGE3)

This routine controls computation of all aerodynamic coefficients, including stability derivatives, for both regular runs and stability derivative runs. If stability derivatives are not requested in the current run, chordwise loading, spanwise loading, and total coefficients are computed by calling STG3FC and STG3FS, respectively, for each fundamental case. Then a summary table of total coefficients for all fundamental cases is printed by calling STG3FT. For each composite case requested, the composite loading routine, STG3C, computes and prints spanwise loading and total aerodynamic coefficients. If stability derivatives are requested, subroutines FUNDER and CØMDER are called, which compute stability derivative coefficients for all fundamental and composite cases, respectively.

Chordwise Loading (STG3FC)

The loading on each element is computed from the vorticity solution produced in the problem solution component program. If requested, the loading is printed, along with detailed expansions of the loading on all singular leading edge and hinge EVD elements.

Spanwise and Total Loading Control (STG3FS)

Spanwise variation of lift, pitching moment and induced drag are computed by calling SLOAD. Total induced drag is also computed by a momentum analysis by calling subroutines SLOADG and TREFTZ. Total lift, pitching moment, and rolling moment are computed in TLOAD, and total induced drag (by pressure integration) and yawing moment are computed by calling TLOADX. If requested, all the above coefficients are then printed. This sequence is repeated for each fundamental case.

Summary Table (STG3FT)

If requested, subroutine STG3FT prints a table of all total aerodynamic coefficients for all fundamental cases. Since no coefficients have been computed for unused fundamental cases, they are initialized to zero before printing.

Composite Case Loading Control (STG3C)

The composite chordwise loading is first computed by superimposing the loading from the required fundamental cases, each multiplied by its respective input scale factor. The loading is also printed, if requested. For composite cases, both the loading at zero angle of attack and the linear variation with angle of attack must be computed. For the nonlinear coefficients, the variation with angle of attack becomes quadratic, and a new "cross-product" term, linear in angle of attack, is also required. The computation of all spanwise and total coefficients corresponding to the zero, linear, quadratic, and cross-product angle of attack terms are computed by calling the various loading utility subroutines. All spanwise and total aerodynamic coefficients are then printed, including a summary table of all total coefficients. Finally, subroutine TABLE is called to give a variation of total lift, drag, and moments

with angle of attack. The entire sequence of computations above are repeated for each composite case requested.

Initialized Data (BLKDTA)

The block data subroutine defines an array of label names for use in labeling the summary tables of total aerodynamic coefficients.

Singular EVD Loading Expansions (EXPLE, EXPH1, EXPH2)

The loading at five points on the EVD element are computed for leading edge singularities and the leading and trailing parts of hinge singularities, respectively.

Spanwise Loading (SLOAD)

This subroutine integrates the chordwise loading at each section to arrive at the sectional values of lift, induced drag and pitching moment. Because of the general nature of the computations, the loading for either fundamental or composite cases can be computed, depending on the element loading given.

Spanwise Cross-Product Loading (SLOADX)

The sectional cross-product values of induced drag are computed by chordwise integration of the loading of each element. This subroutine is utilized only in computation of composite cases.

Spanwise Vorticity (SLOADG)

This subroutine integrates the chordwise loading at each section to get the total vorticity of the jet wing system for use in calculation of induced drag by a momentum analysis. Integration is from the leading edge to the trailing edge for unblown sections, and from the leading edge to infinity for sections with a jet.

Total Loading (TLOAD)

This subroutine performs spanwise integration of the sectional values of lift, jet strength, and pitching moment to produce total lift, induced drag, jet strength, and pitching and rolling moments. Only total coefficients for fundamental cases are computed.

Total Loading (TLOADO)

This subroutine computes total lift and pitching and rolling moments for all composite cases at zero angle of attack. The coefficients are computed by summing the fundamental case coefficients, each multiplied by its respective input scale factor.

Total Loading (TLOADX)

This subroutine computes total induced drag (both momentum wing pressure integral methods) and yawing moment coefficients at zero angle of attack by spanwise integration of the appropriate sectional data. Both fundamental and composite case coefficients may be computed.

Trefftz Plane Downwash (TREFTZ)

This subroutine computes the induced downwash at the Trefftz Plane (infinity) due to all the loading of the complete jet-wing system, for either fundamental or composite cases. These data are used in the momentum induced drag method.

Composite Case Summary (TABLE)

This subroutine computes and prints a table of the variation of lift, induced drag (momentum method), and pitching, yawing, and rolling moments with angle of attack. The printout is given for each composite case.

Fundamental Case Stability Derivatives (FUNDER)

This subroutine controls calculation of stability derivatives for all fundamental cases. Subroutine STG3FC is first called to generate the chordwise loading for the stability fundamental cases. Then both chordwise and

spanwise integration are performed by calling subroutines SUMIT1 and SUMIT2, to yield stability derivatives due to rolling and yawing rates. For the last fundamental case (due to rolling rate) certain stability derivatives are derived directly from chordwise and spanwise loading in the normal manner by calling STG3FS.

Composite Case Stability Derivatives (COMDER)

This subroutine controls computation of stability derivatives for all composite cases. First the composite case loading is assembled, requiring reading of the fundamental case solutions for the first run from mass storage Unit 1. Then the subroutines SUMIT1 and SUMIT2 are called for chordwise and spanwise integration of the loading to obtain the required stability derivatives. Finally a table is printed showing the components of the stability derivatives and the variation of the derivatives with angle of attack.

Stability Derivative Integration (SUMIT1)

This subroutine computes the derivatives of yawing moment due to rolling, and rolling moment due to yawing by chordwise and spanwise integration of the loading on each element.

Stability Derivative Integration (SUMIT2)

This subroutine computes the derivative of yawing moment due to yawing by chordwise and spanwise integration of the loading on each element.

Stability Derivative Table (STABLE)

This subroutine prints, for each composite case, a summary of all stability derivatives and the terms used to compute each. An angle of attack table is then computed and printed, containing all the stability derivatives which depend on angle of attack.

Control of Utility Routines for Stability Runs (STAGE4)

During the first run of a stability derivative run sequence, an extra

fundamental case is created with induced camber intended to simulate a rate of pitching. This is accomplished by calling subroutine BCPICH, and all the resulting geometric data is printed by calling OUT2. During the second run of a stability derivative run sequence, the fundamental case solutions from the first run are saved on mass storage Unit 1 in subroutine SAVECP. The fundamental case geometry is then redefined, with induced camber simulating rates of yawing and rolling, by subroutines BCYAW and BCROLL, respectively. Again, OUT2 gives a record of the new fundamental case geometry.

Fundamental Case Output (OUT2)

This subroutine is the same as subroutine OUT2, and prints the geometric data defining the stability derivative fundamental cases.

Save First Run Solution (SAVECP)

For stability derivative runs, the loading solution from the first run must be saved for use in the second run. This is accomplished by writing the data on mass storage Unit 1, directly behind the matrix information previously stored.

Pitching Rate Induced Camber (BCPICH)

This subroutine defines the induced camber angles on all wing and jet EVD elements, which result from simulation of the wing pitching about the input center of gravity location.

Rolling Rate Induced Camber (BCROLL)

This subroutine defines the induced camber angles on all wing and jet EVD elements, which result from simulation of the wing rolling about the x-axis.

Yawing Rate Induced Camber (BCYAW)

This subroutine defines the induced camber angles on all wing and jet EVD elements, which result from simulation of the wing yawing about the z-axis.

3.3 EVD Functions-Numerical Restrictions

Because of the characteristic of all digital computers of defining numbers to only a limited number of significant digits, it sometimes happens that equations cannot be evaluated with the required accuracy. This is particularly a problem where the difference of two numbers of nearly the same magnitude is taken. As the two numbers approach each other, the difference loses accuracy, and if they are identical (to the number of digits the computer can hold), the difference loses all accuracy.

For the four EVD downwash influence functions (see Appendix I of Reference 1), the above type of problem has been encountered in their evaluation when the control point is in the far-field or, for certain unique locations, in the near-field region of any EVD element. These can generally occur along the leading and trailing edges of any EVD element, and at the apex of a Regular EVD.

The following restrictions and approximations have been adopted, without loss of generality, in order to prevent random inaccuracies from affecting the results. In the case of far-field cutoff, the true limiting values of each function are used, and the only approximations are that these limiting values are used somewhat "closer" than real infinity, where they correctly apply. For near-field cutoff, the limiting values are used in small regions near the points where the values correctly apply. The subscript i refers to the control point at which the downwash is being computed, due to the vorticity of element j.

EVD Influence Function	Region	Value Used
Regular*	$\frac{x_i-x_i}{\frac{1}{2}\left(\delta_j+\delta_{j-1}\right)} > 100$	$a\left(\infty,\left(y_i-y_j\right)\right)=-\frac{1}{4\pi}\left(\delta_j+\delta_{j-1}\right)\left(\frac{1}{y_i-y_j-\Delta}-\frac{1}{y_i-y_j+\Delta}\right)$
	$x_i-x_j = 0,\ y=0$	Not encountered
	$x_i-x_j = -\delta_{j-1},\ y=0$	Not encountered
	$x_i-x_j = \delta_j,\ y=0$	Not encountered

*Reference 1: Appendix I, Section I.1

EVD Function	Region	Value Used
Leading* Edge	$\left(x_i-x_j\right)/\delta_j > 100$	$a\left(\infty,\left(y_i-y_j\right)\right) = -\frac{1}{2\pi}\left(\frac{1}{y_i-y_j-\Delta} - \frac{1}{y_i-y_j+\Delta}\right)$
	$\left\lvert\left(x_i-x_j\right)/\delta_j\right\rvert < 10^{-4}$; $\left\lvert\left[\left(x_i-x_j\right)/\delta_j\right] - 1\right\rvert < 10^{-6}$	$a\left(\left(x_i-x_j\right),\left(y_i-y_j\right)\right) = a\left(0,\left(y_i-y_j\right)\right)$
Infinity+	$\left(\frac{x_i-x_j+d}{y_i-y_j-\Delta}\right) > 10^6$	$a\left(\left(x_i-x_j\right),\left(y_i-y_j\right)\right) = -\frac{1}{2\pi}\left(\frac{\delta_{j-1}}{2} + d\right)\left(\frac{1}{y_i-y_j-\Delta} - \frac{1}{y_i-y_j+\Delta}\right)$
	$\left\lvert\frac{x_i-x_j+d}{d}\right\rvert < 10^{-2}$	$a\left(\left(x_i-x_j\right),\left(y_i-y_j\right)\right) = a\left(-d,\left(y_i-y_j\right)\right)$
Hinge‡ (Singular Part)	$\left\lvert\frac{x_i-x_j}{\frac{1}{2}\left(\delta_j+\delta_{j-1}\right)}\right\rvert > 7.5$	$b\left(\left(x_i-x_j\right),\left(y_i-y_j\right)\right) = \frac{\beta}{\pi^2}\left[-\left(\delta_{j-1}+\delta_j\right) - \left(\delta_{j-1}\log\delta_{j-1}+\delta_j\log\delta_j\right)\right]\left[\frac{1}{y_i-y_j-\Delta} - \frac{1}{y_i-y_j+\Delta}\right]$
	$\left\lvert\frac{x_i-x_j}{\frac{1}{2}\left(\delta_j+\delta_{j-1}\right)}\right\rvert < 10^{-4}$	$b\left(\left(x_i-x_j\right),\left(y_i-y_j\right)\right) = b\left(0,\left(y_i-y_j\right)\right)$
	$\left\lvert\frac{x_i-x_j+\delta_{j-1}}{\delta_{j-1}}\right\rvert < 10^{-6}$	$b\left(\left(x_i-x_j\right),\left(y_i-y_j\right)\right) = b\left(-\delta_{j-1},\left(y_i-y_j\right)\right)$
	$\left\lvert\frac{x_i-x_j-\delta_j}{\delta_j}\right\rvert < 10^{-6}$	$b\left(\left(x_i-x_j\right),\left(y_i-y_j\right)\right) = b\left(\delta_j,\left(y_i-y_j\right)\right)$
	$\left\lvert\frac{\delta_{j-1}}{\lvert x_i-x_j\rvert} - 1\right\rvert < 10^{-4}$	$b\left(\left(x_i-x_j\right),\left(y_i-y_j\right)\right) = b\left(\delta_{j-1},\left(y_i-y_j\right)\right)$
	$\left\lvert\frac{\delta_j}{\lvert x_i-x_j\rvert} - 1\right\rvert < 10^{-4}$	$b\left(\left(x_i-x_j\right),\left(y_i-y_j\right)\right) = b\left(\delta_j,\left(y_i-y_j\right)\right)$

* Reference 1: Appendix I, Section I 2
\+ Reference 1: Appendix I, Section I 3
‡ Reference 1: Appendix I, Section I 4

EVD Function	Region	Value Used
Hinge‡ (Singular Part, cont'd)	$\left\vert \frac{\delta_j - x_i - x_j}{\delta_j} \right\vert < 10^{-6}$	$b\left((x_i - x_j), (y_i - y_j)\right) = b\left(\delta_j, (y_i - y_j)\right)$
	$\left\vert \frac{x_i - x_j + \delta_{j-1}}{\delta_{j-1}} \right\vert < 10^{-6}$	$b\left((x_i - x_j), (y_i - y_j)\right) = b\left(-\delta_{j-1}, (y_i - y_j)\right)$

Note: Values of Hinge EVD regular part are equivalent to Regular EVD.

‡ Reference 1: Appendix I, Section I 4

4.0 INPUT DATA INSTRUCTIONS

The program input consists of various types of cards containing all the information required to define a particular problem. The number of cards of each type may vary from problem to problem, but the sequence is the same in all runs. As a result of options chosen by the user, some types of cards may not be required, but the sequence of the remaining cards must not be changed. Load sheets for all cards are shown in Appendix D. The input deck is shown diagrammatically in Figure 6.

For symmetric and anti-symmetric wings, all input and output data are for the right half of the wing only. Figure 7 identifies some of the planform coordinates required as input. These will be discussed in detail below. Sectional input normally begins at the right wing tip section, working inboard section by section. However, most data required for description of the jet is only needed at sections where the jet exists, and unblown sections should be skipped as indicated in the instructions below.

Figure 8 shows the reference coordinate system used for both input and output data. This is a wind axis system with the freestream direction always aligned with the x-axis. It must be noted that if a wing is to be analyzed in the yawed position, it must be input as a non-symmetric planform, rotated about the wing apex (z-axis) with respect to the x-axis. All wing sections will still be aligned with the x-axis, including the wing tip sections. The printed output will also be referenced to the wind axis system, and the user must make suitable transformations if coefficients are required in a body axis system.

In the input description which follows, all input values are real floating point numbers (F format), unless otherwise specified. All specified integers should be right-justified in their fields.

4.1 Input Description

- Title Card - This card provides any desired description of the computer run. The title will be printed at the top of the first page of output.

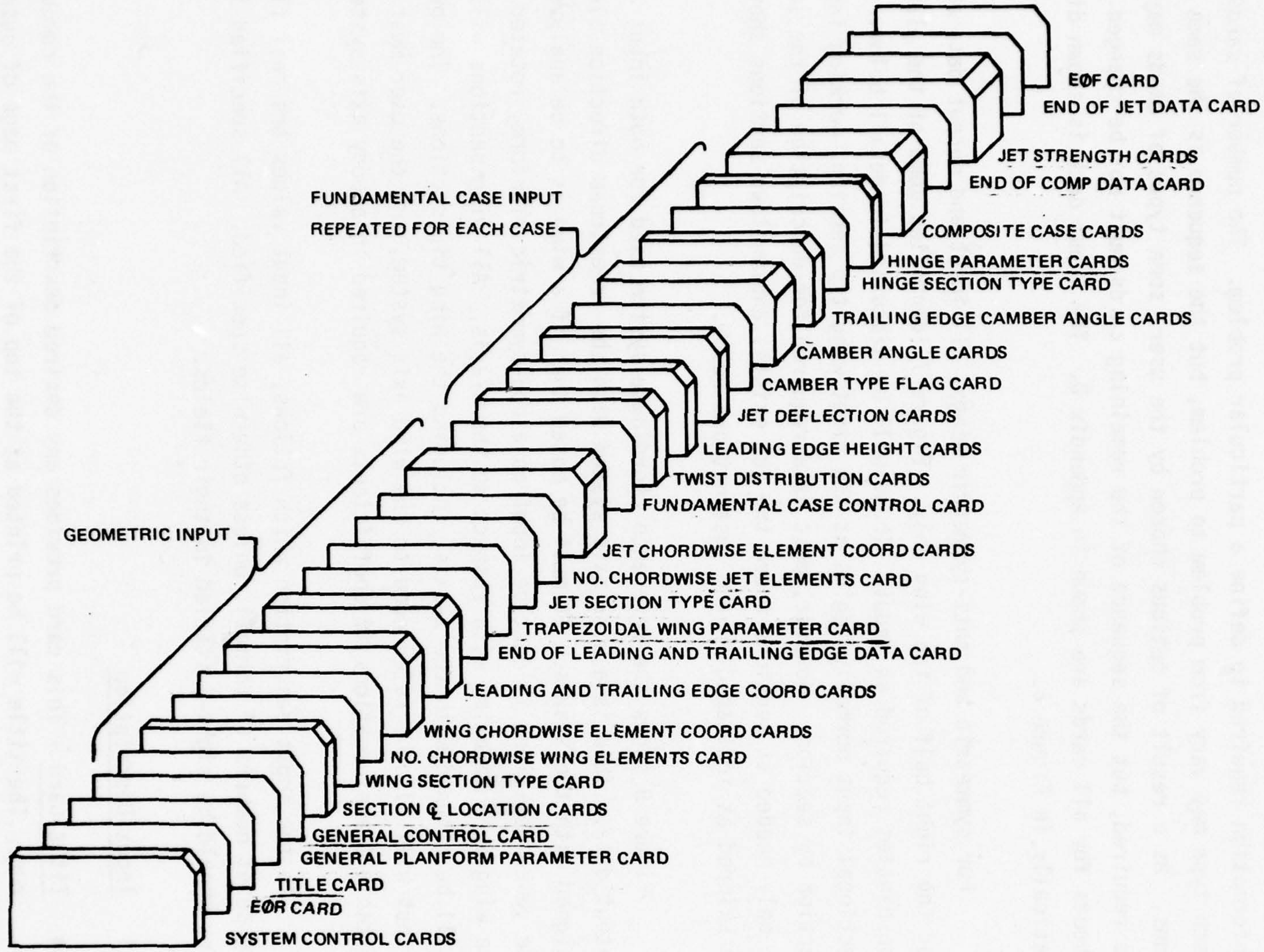


FIGURE 6. ARRANGEMENT OF INPUT CARD DECK

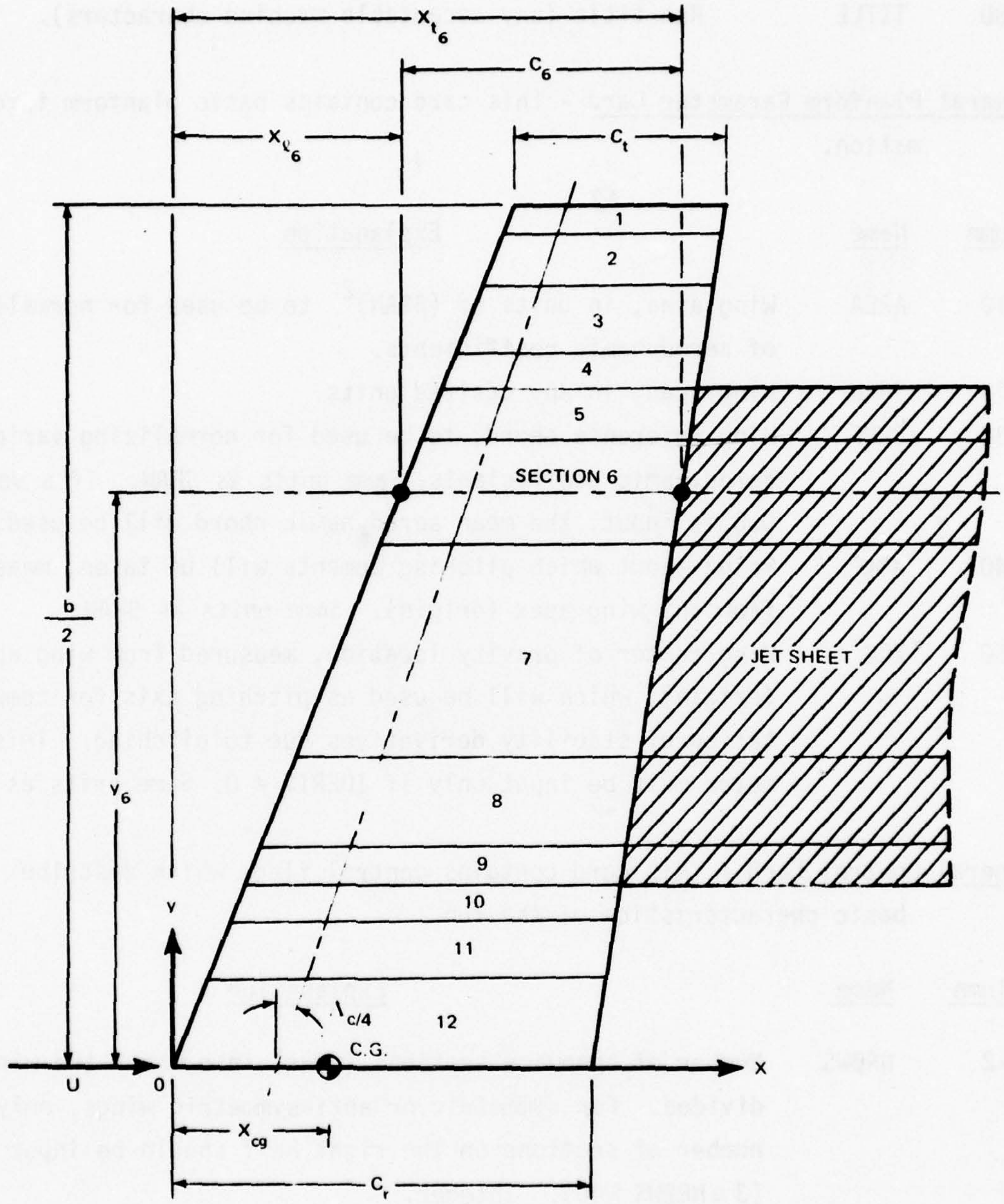


FIGURE 7. PROGRAM PLANFORM DIMENSIONS AND NOTATION

Column	Name	Explanation
1-80	TITLE	Run title (any acceptable machine characters).

● General Planform Parameter Card - This card contains basic planform information.

Column	Name	Explanation
1-10	AREA	Wing area, in units of $(SPAN)^2$ to be used for normalization of aerodynamic coefficients.
11-20	SPAN	Wing span, in any desired units.
21-30	CREF	Wing reference chord, to be used for normalizing various aerodynamic coefficients. Same units as SPAN. If a value of 0.0 is input, the mean aerodynamic chord will be used.
31-40	XMC	Point about which pitching moments will be taken, measured from the wing apex (origin). Same units as SPAN.
41-50	XCG	Wing center of gravity location, measured from wing apex (origin), which will be used as pitching axis for computation of stability derivatives due to pitching. This parameter must be input only if IDERIV $\neq$ 0. Same units as SPAN.

● General Control Card - This card contains control flags which describe the basic characteristics of the run.

Column	Name	Explanation
1-2	NRØWS	Number of spanwise sections (rows) into which the wing is divided. For symmetric or anti-symmetric wings, only the number of sections on the right half should be input ($3 \leq$ NRØWS ≤ 40). Integer.
3-4	NCASES	Total number of fundamental cases. It must be noted that the angle of attack case is always set up automatically as fundamental case number one, and no input data is required for it. Therefore, NCASES must be one more than the number of cases for which input data will be given, to allow for the angle of attack case. ($1 \leq$ NCASES ≤ 10). Integer.

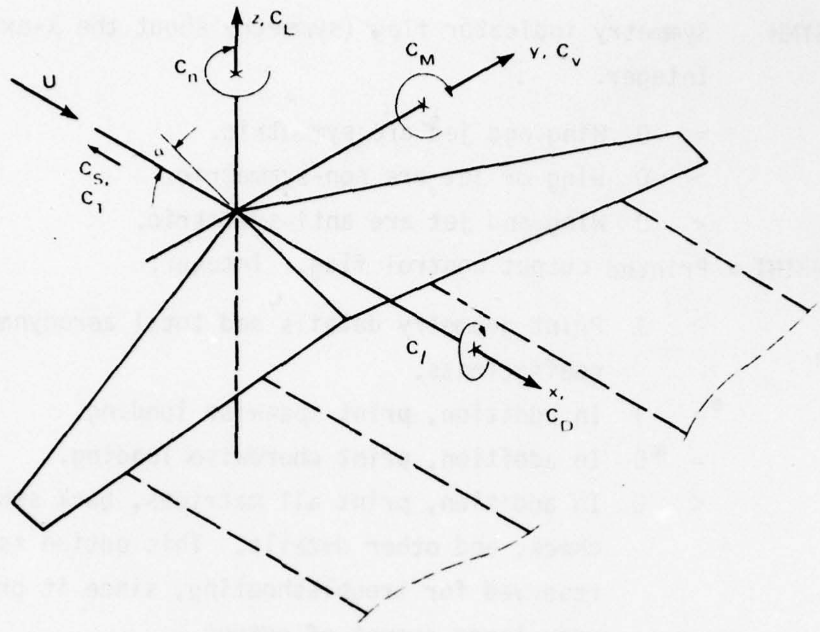


FIGURE 8a. REFERENCE COORDINATE SYSTEM FOR AERODYNAMIC COEFFICIENTS

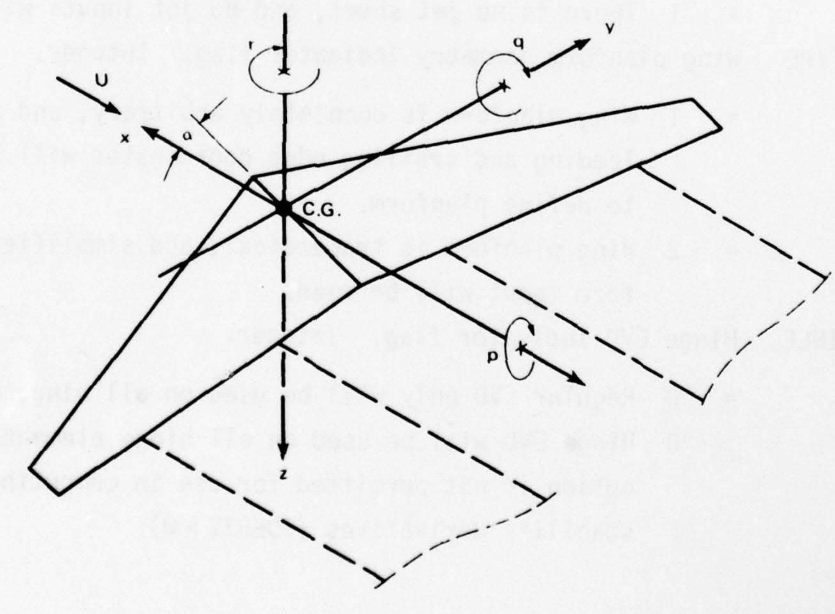


FIGURE 8b. REFERENCE COORDINATE SYSTEM FOR STABILITY DERIVATIVES

FIGURE 8. REFERENCE COORDINATE SYSTEM

Column	Name	Explanation
5-6	ISYMM	Symmetry indicator flag (symmetry about the x-axis). Integer. = 0 Wing and jet are symmetric. > 0 Wing or jet are non-symmetric. < 0 Wing and jet are anti-symmetric.
7-8	IPRINT	Printed output control flag. Integer. > 1 Print geometry details and total aerodynamic coefficients. = 1 In addition, print spanwise loading. = 0 In addition, print chordwise loading. < 0 In addition, print all matrices, back substitution check, and other details. This option is normally reserved for troubleshooting, since it produces a very large amount of output.
10	JETFLG	Jet indicator flag. Integer. = 0 There is a jet sheet. = 1 There is no jet sheet, and no jet inputs will be read.
12	IGTYPE	Wing planform geometry indicator flag. Integer. = 1 Wing planform is completely arbitrary, and sectional leading and trailing edge coordinates will be read to define planform. = 2 Wing planform is trapezoidal, and simplified planform input will be read.
14	IHINGE	Hinge EVD indicator flag. Integer. = 0 Regular EVD only will be used on all hinge elements. > 0 Hinge EVD will be used on all hinge elements. This option is not permitted for use in computing dynamic stability derivatives (IDERIV > 0).

Column	Name	Explanation
16	IDERIV	Dynamic stability derivative control flag. Integer. = 0 A basic run will be executed, with no stability derivatives computed. > 0 A basic run will be executed, and in addition, a dynamic stability derivative run will be executed. This option requires approximately double the basic computing time.

- Section Centerline Location Cards - These cards contain the spanwise locations of the centerline of each wing (and jet)section. Eight values per card, maximum of five cards (40 sections) allowed.

Column	Name	Explanation
1-10,11-20, etc.	Y	Spanwise distance from wing centerline (x-axis) to section centerline, normalized by SPAN/2. All values must be $(-1.0 < Y < 1.0)$. NRØWS values must be input, beginning at the right wing tip and working to the following: (a) Wing centerline, for symmetric or anti-symmetric wings. (b) Left wing tip, for non-symmetric wings.

- Wing Section Type Card - This card indicates the chordwise arrangement of EVD elements for each section on the wing.

Column	Name	Explanation
1-2, 3-4, etc.	ICTYPE	Type number of each wing section. The arrangement of chordwise EVD elements in a row (both number of elements and the x/c location of each) determines the wing row type. Any two sections which have the same number and x/c locations of all EVD elements, have the same ICTYPE value. NRØWS values must be input. There may not be more than ten different values (i.e., section types). The highest value input is called NWTYPE, and all values less than NWTYPE

Column	Name	Explanation
		must be used, that is, no "gaps" are allowed in the sequence 1 through NWTYPE. Integer.

- Number of Chordwise Wing Elements - This card contains the number of chordwise EVD elements for each wing section type.

Column	Name	Explanation
1-2,3-4, etc. Up to CC10	NI	Number of chordwise EVD elements of each wing section type. There must be NWTYPE values ($1 \leq NWTYPE \leq 10$) in the sequence for section type 1, 2, 3, ...(i.e., ascending order of row types). At least two but not more than twenty chordwise EVD elements must be input on each wing section. ($2 \leq NI \leq 20$). Integer.

- Wing Chordwise Element Coordinates - These cards contain the x/c coordinates of each EVD element of each section type. NWTYPE sets of cards required, each with NI values of x/c. Maximum of ten sets, three cards per set.

Column	Name	Explanation
1-10,11-20, etc.	XB	The chordwise coordinate of each EVD vortex point, measured from the sectional leading edge and normalized by the sectional chord. The vortex point is defined as the leading edge for all Leading edge EVD's and the "peak" point for all Regular and Hinge EVD's. The first value of each set must be 0.0 (leading edge) and the last value must be less than 1.0. A maximum of 20 values (i.e., 20 EVD elements) is permitted for each set, but the total number of elements on the wing and jet combined must not exceed 600.

- **Leading and Trailing Edge Coordinates** - These cards contain the coordinates of the leading and trailing edges of the given sections. Only stations on each side of breaks in the leading or trailing edges need be input. The tip and root sections must be input. The leading and trailing edges are interpolated for sections not given by putting straight lines between those sections which are given. This method of planform definition is used only if IGTYPE = 1. Number of cards required is ≥2 and ≤NRØWS.

Column	Name	Explanation
1-10	Y	Spanwise distance from a section centerline to the centerline of the wing (x-axis), normalized by the half span. Each value must be the same as one of those already input on the section centerline location cards. Values on the right wing half are positive, and values on the left wing half (nonsymmetric only) are negative.
11-20	XLEAD	Chordwise distance from section leading edge (at section centerline) to the wing apex (y-axis). Same units as SPAN.
21-30	XTRAIL	Chordwise distance from section trailing edge (at section centerline) to the wing apex (y-axis). Same units as SPAN.
1	9	A 9 must appear in CC1 of the next card after all the above coordinate cards to signal that all desired sections have been input. This card is required only if IGTYPE = 1.

- **Trapezoidal Wing Parameters** - This card contains planform information for simple trapezoidal wings. It replaces the above coordinate cards including the 9 card when IGTYPE = 2. This type of input may be used only when the wing planform is symmetric.

Card Column	Variable Name	Explanation
1-10	ARATIØ	Wing aspect ratio, $(SPAN^2/AREA)$.
11-20	SWEEP	Sweep angle of wing quarter-chord line, in degrees.
21-30	TR	Wing taper ratio, $\left(CHØRD_{\text{wing tip}} / CHØRD_{\text{axis of symmetry}} \right)$

- Jet Row Type Card - This card indicates the chordwise arrangement of EVD elements for each section on the jet sheet. Required only if JETFLG = 0 .

Column	Name	Explanation
1-2,3-4, etc.	IJTYPE	The type number of each jet section. This variable is similar to the ICTYPE variable, except that for sections with no jet a value of 0 must be input. The number of different jet section types is called NJTYPE. The number of nonzero values input is NROWSJ, the number of sections having a jet. Integer.

- Number of Chordwise Jet Elements - This card contains the number of chordwise EVD elements for each jet section type. Required only if JETFLG = 0 .

Column	Name	Explanation
1-2, 3-4, etc.	NI	Number of chordwise EVD elements of each jet section type. This variable is similar to NI for wing sections above, except that there must be NJTYPE sections. At least 2 but not more than 10 chordwise EVD elements must be input on each jet section ($2 \leq NI \leq 10$). Integer.

- Jet Chordwise Element Coordinates - These cards contain the x/c coordinates of each element of each jet section type. NJTYPE sets of cards required, each with NI values of x/c. Maximum of 10 sets, 2 cards per set. Required only if JETFLG = 0.

Column	Name	Explanation
1-10,11-20, etc.	XB	The chordwise coordinate of each EVD vortex point, measured from the sectional leading edge (at centerline) and normalized by the sectional chord. The first value of each set must be 1.0 (trailing edge) and a maximum of 10 values is permitted for each set ($2 \leq NI \leq 10$).

- Fundamental Case Control Card - This card identifies the types of linear geometric variations to be included in each fundamental case. The number of fundamental cases input must be one less than NCASES, to allow for the angle of attack case. A new fundamental case control card is required for each of the (NCASES - 1) input cases. In each of the flags below, a zero value indicates omission of the respective type of input for that fundamental case. A non-zero value indicates the variation will be included and input must be given to define it.

Column	Name	Explanation
2	INTWST	Spanwise twist distribution flag. Integer.
4	INHITE	Leading edge vertical displacement flag. Integer.
6	INDELJ	Jet deflection flag. Integer.
8	INCAMB	Camber flag. Integer.
10	INBETA	Wing hinge deflection flag. Integer.

- Twist Distribution Card - These cards contain the spanwise distribution of wing twist. NRØWS values required, eight per card. Required only if INTWST ≠ 0.

Column	Name	Explanation
1-10,11-20 etc.	TWIST	Sectional wing twist, in degrees, at section centerline, with respect to the wing reference plane. Positive is in the same sense as a positive angle of attack. See Figure 3b ②.

- Leading Edge Height Card - These cards contain the displacement of the wing leading edge from the wing reference plane. NRØWS values required, eight per card. Required only if INHITE ≠ 0.

Column	Name	Explanation
1-10,11-20, etc.	HO	Displacement coordinate of sectional leading edge from the wing reference plane, normalized by the sectional chord. These values are used only for computation of the moment arm of the jet reaction thrust contribution to pitching moments. Leading edge displacement may be the result of dihedral, twist, nonlinear movement of a leading edge device, etc. Translation resulting from ordinary linear leading and trailing flap deflections and angle of attack are accounted for automatically by the program.

- Jet Deflection Card - These cards contain the spanwise variation of jet deflection, relative to the trailing edge. NRØWSJ values are required, eight per card. Required only if INDELJ ≠ 0.

Column	Name	Explanation
1-10,11-20, etc.	DJ	Jet turning angle, in degrees, relative to the mean line of the trailing edge. Positive deflection is downward. See Figure 3b ⑨.

- Camber Type Flag Card - This card indicates the chordwise distribution of camber for each section on the wing. Required only if INCAMB ≠ 0 .

Column	Name	Explanation
1-2,3-4 etc.	ICT	Camber type number of each wing section. The arrangement

of camber, including all positions (x/c) and angles must be the same in order for two sections to have the same value of ICT. NRØWS values must be input, but a maximum of 10 different types is allowed. The highest value input is called NCT, and there may be no "gaps" in the numbering sequence between 0 and NCT. A zero value indicates no camber. Integer.

- Camber Angle Cards - These cards contain the camber angles for each camber section type. NCT sets of cards required, three cards per set maximum. Required only if INCAMB ≠ 0.

Column	Name	Explanation
1-10,11-20, etc.	AC	The camber angle, in degrees, at the downwash control point of each EVD element (i.e., the point arbitrarily chosen as halfway between any two adjacent XB points, including the trailing edge). Positive values are in the same sense as positive angles of attack. See Figure 3b ③.

- Trailing Edge Camber Angle Card - These cards contain the trailing edge angle due to camber only. NRØWSJ values are required. Required only if both INCAMB ≠ 0 and JETFLG = 0.

Column	Name	Explanation
1-10,11-20 etc.	ACTE	Trailing edge angle, in degrees, due to camber only. These values are required only for sections which have a trailing jet, and are used only in computing the total jet deflection angle with respect to the freestream.

- Hinge Section Type Card - This card identifies the arrangement of leading and trailing flap hinges on each section. NRØWS values are required. Required only if INBETA ≠ 0.

Column	Name	Explanation
1-2,3-4 etc.	IHT	Hinge section type flag. If there are no hinges, a value of 0 should be input. Sections with the same hinge section type must be alike in number of hinges, their location (x/c), their type (leading or trailing flap) and in all deflections. There may be as many as NRØWS different section types. The number of different types is called NHT, and there may be no gaps in the sequence 0 through NHT. Integer.

- Hinge Parameter Cards - These cards contain the location, type (leading or trailing edge flap) and turning angle at each hinge on a given type of hinge section. NHT cards required. Each section may have a maximum of four hinges. Required only if INBETA ≠ 0.

Column	Name	Explanation
1-10,21-30 etc.	XBH	Distance from the sectional leading edge to the hinge point, normalized by the sectional chord. This value must be the same as one of the XB coordinates input for the section.
11,31, etc.	ILT	Hinge type identification flag. Integer. = 0 Trailing flap hinge (positive deflection in the sense of positive angle of attack). ≠ 0 Leading flap hinge (positive deflection in the sense of negative angle of attack).
12-20,32-40, etc.	BETA	Hinge deflection angle, in degrees, of the element behind the hinge point, relative to the element before the hinge point. See Figure 3b ④, ⑥, ⑦.

- Composite Case Cards - These cards specify the desired superposition of the linear fundamental cases. A maximum of 24 composite cases may be requested. No composite cases may also be specified (9 card alone required). See Figure 3c.

Column	Name	Explanation
1-2,9-10, etc.	N	Fundamental case number which is to be included in forming a given composite case. Integer.
3-8,11-16, etc.	A	Multiplication factor to be applied to the fundamental case identified in the previous two card columns.
1	9	A "9" must appear in CC1 of the card following the composite case cards to signal the end of composite case input. (Required even if there are no composite case cards input).

- Jet Strength Cards - These cards contain the jet strength for all sections which have a jet. NROWSJ values are required. An unlimited number of sets of cards may be input, maximum of five cards per set. Required only if JETFLG = 0 .

Column	Name	Explanation
1-10,11-20, etc.	CMU	Sectional jet momentum strength for each jet row. CMU is defined as CMU = J/qc(y), where J is the sectional jet momentum per unit span and q and c are the dynamic pressure and sectional chord, respectively. There may be no zero values input, unless all values are zero (indicating no jet at all).
1	9	A "9" must appear in CC1 of the card following all CMU cards to signal the end of CMU input. Required only if JETFLG = 0 .

4.2 Input Restrictions

Most of the limitations imposed on the range of input values are designed to minimize computer storage requirements, but still to provide the capability for adequate treatment of complicated configurations. While all of the input restrictions have been stated or implied in the input description section above, they will be summarized here for convenient reference.

1. $3 \le NR\O WS \le 40$
2. $1 \le NCASES \le 10$
3. IGTYPE = 1 or 2
4. Number of chordwise wing elements on any section $2 \le NI \le 20$
5. Number of chordwise jet elements on any section $2 \le NI \le 10$
6. Maximum of ten different section types each on wing and jet
7. On wing $0.0 \le XB < 1.0$
8. On jet $1.0 \le XB$
9. Only NROWSJ values required for DJ, ACTE and CMU input
10. $0.0 \le CMU < 800.0$
11. For stability derivative runs, only one set of CMU data may be input

Due to certain curve fitting restrictions in the calculation of induced drag by the momentum method, at least three adjacent sections must be used in defining either a jet sheet or a region on the wing which has no trailing jet sheet. Thus, a jet cannot consist of only one or two sections. Likewise, inboard or outboard of any jet, there must be at least three unblown wing sections. If the three-in-a-row rule is not followed, the program will stop automatically.

5.0 OUTPUT DATA DESCRIPTION

The program output consists entirely of data written to the standard print Unit 6. Four print options, controlled by the input value of the IPRINT flag described above, are available to the user. The normal printout is given for the default value of zero, and includes chordwise and spanwise loading and total wing aerodynamic coefficients. If the details of chordwise or spanwise loading are not required, a large part of the print and a slight amount of computing time may be saved by requesting one of the more restricted forms of output.

If more information is required for program checkout or troubleshooting, the extended print option may be used. Under this option, all of the pertinent data of the problem solution are printed, including downwash coefficients, augmented matrix coefficients, vorticity solution and back substitution check for recovery of the right side matrix. The matrix information is given in a row by row sequence, each row corresponding to one EVD element. For each row the influence coefficients of all EVD elements on the one in question are listed across the page. For example, the downwash influence due to Element 12 on Element 6 would be the second number of the second line listed under matrix Row 6.

The detailed print option, when used in a stability derivative run, will also give for the second run the chordwise loading, spanwise loading of the last fundamental case, and the summary table of total coefficients for all fundamental cases. These data are not intended for direct use, but mainly for indirect checking of the consistency of certain stability derivative results.

It will be helpful for the user to refer to the sample problem output of Appendix A during discussion of the output data given below. All output symbols and labels are listed in the nomenclature section and all angles are measured in degrees.

Even though many significant digits are printed, this is by no means meant to imply that the coefficients are accurate to such a degree. Six or eight significant digits are printed only so that the effects due to small

deflections or planform changes will be visible in the F format. While changes in the last significant digits may indicate the trends of effects due to small changes, they must not be construed to indicate their magnitude accurately.

The composite case linear aerodynamic characteristics are lift, pitching moment, root bending moment, and rolling moment. They are composed of a value at zero angle of attack and a linear variation with angle of attack, and may be calculated as follows:

$$C_L = C_{L_{\alpha=0}} + C_{L_\alpha} \alpha$$

(CCL = CCLO + CCLA * ALPHA)

The nonlinear characteristics include all of the drag contributions, yawing moment and side force, and are composed of a value at zero angle of attack, a square variation with angle of attack, and a "cross-product" linear variation with angle of attack as follows:

$$C_D = C_{D_{\alpha=0}} + C_{D_\alpha} \alpha + C_{D_{\alpha^2}} \alpha^2$$

(CCD = CCDO + CCDX * ALPHA + CCDA2 * ALPHA ** 2)

The proper methods of composition of the dynamic stability derivatives vary greatly, and are shown in the printout for each of the types of derivatives.

As noted in the input data instructions, the reference system of both input and output data is a wind axis system, with the x-axis aligned with the freestream. If it is required to analyze a yawed wing, the wing must be input in the yawed position and considered as non-symmetric. If aerodynamic coefficients are required in a body axis system, they must be transformed by the user. In the linearized analysis, the only contribution to side force should be the leading edge suction, taken normal to the local leading edge.

The program automatically normalizes all the geometry data by dividing by half the input value of the wing span (i.e. SPAN). Thus, the program works

exclusively with a wing which has a span of two units, and all output is for this normalized wing.

5.1 Wing Characteristics and Control Flags

The program title header and the input case title are printed for reference. Next the major wing planform characteristics and input control flags are printed. Since the program will automatically redefine some of the parameters and flags under certain run conditions, both the input and internally used values are printed for reference.

5.2 Fundamental Case Geometry

The page header is printed at the beginning of data for each fundamental case. For each section, the pertinent dimensions are given on the first line. Next the detailed data for the wing are given, with the general information printed on the first line, followed by the chordwise arrangement of all element locations, angles and EVD types. Each number printed across the page corresponds to one EVD element, beginning at the sectional leading edge and working toward the trailing edge. The data given include, for each element, the following:

a. Chordwise location, first normalized by the local chord, then in dimensional units.
b. Chordwise element length.
c. Total incidence angle resulting from all forms of fundamental case deflections.
d. Hinge turning angle of each element.
e. EVD type.

Next, the jet element data is given, if there is a jet at that section. The first line contains data applying to the whole jet section, and is followed by the chordwise arrangement of EVD element data. The element incidence angle is not printed since it has no physical meaning, and the hinge turning angles must be zero for all but the leading jet element. The length of the jet infinity EVD element will always be printed as a large number overflowing the format, since it is theoretically infinite and is automatically assigned the value 10^{10}.

The values of TYPE identify the following EVD types:

10	Regular EVD	
20	Leading Edge EVD	
30	Infinity EVD	
41	Leading edge flap hinge	Hinge EVD
42	Trailing edge flap hinge	Hinge EVD
43	Trailing edge - jet-emersion	Hinge EVD

For all fundamental cases after the first the printout of element dimensional locations and lengths are skipped, since they are the same for all cases. The normalized chordwise locations are also repetitious, but are printed for reference in each case.

5.3 Sectional Jet Strength

This output is given for reference and includes the c_μ value at every section, whether or not it is a jet section. This printout is skipped if JETFLG $\neq$ 0.

For dynamic stability derivative runs, an extra fundamental case is created automatically by the program, with incidence angles defined according to the quasi-steady induced camber due to a one degree/second pitching rate about the XCG location.

5.4 Fundamental Case Coefficients

After the problem solution is completed, the loading, ΔC_p, on each EVD element is printed for all fundamental cases. These data are given section by section, with the wing element loading first and the jet element loading second. In addition, detailed loading at small intervals are given for all leading edge and hinge EVD elements. This is intended primarily as an aid to the user in plotting the chordwise loading. The hinge detailed print is given only if IHINGE $\neq$ 0, since hinge loadings are otherwise represented by regular EVD elements rather than singular hinge EVD elements.

The spanwise loading data include, for each fundamental case, the lift,

induced drag, pitching moment, and center of lift for each wing section. At the bottom of each column, the values labeled "TOTAL" are the corresponding integrated aerodynamic coefficients for the whole wing. The sectional pitching moment coefficients are taken about the sectional leading edges, while the total coefficients are taken about both the wing apex (origin) and about the input XMC location. Sectional center of pressure and center of lift data are all normalized by the sectional chord. The total values indicate the chordwise centers of pressure and lift, each normalized by either reference chord or wing semispan. The total value of induced drag computed by the momentum method is given under the column headed ALFINF.

For dynamic stability derivative runs, the derivatives due to pitching are derived from the last fundamental case and are printed below the spanwise loading data.

Finally, a summary table of all total aerodynamic coefficients for all fundamental cases is printed. These values are the same as those printed under the spanwise loading data, but also include rolling and yawing moments. The most important coefficients are labeled with asterisks.

It should be noted that in order to compute the sectional pitching moment about the leading edge due to the thrust component (in the freestream direction) of jet reaction acting at the trailing edge, the vertical distance between the leading and trailing edges must be known. For wings with flap or slat fundamental cases, the program computes this distance using the hinge points and respective deflection angles. If, however, the same deflections are described by a camber fundamental case, the program has no way of knowing that the leading or trailing edges are displaced, and therefore, the jet thrust pitching moment contribution will always be zero. The user is thus warned to use care when interpreting the pitching moment results for jet-wings where flap or slat deflections are treated as camber alone.

5.5 Composite Case Coefficients

The composite case output is similar to that for fundamental cases, except that all data is given for each composite case before going on to the

next. The first line of data contains the input multiplication factors for the fundamental cases comprising this particular composite case. Then the chordwise loading is printed for each section. The Δc_p values are for zero angle of attack, labeled (A = 0), and for the variation of loading with angle of attack, labeled (A = 1). The fundamental case factors apply only to the loading at zero angle of attack, and the loading variation with angle of attack is taken directly from fundamental case number one.

As noted above, the CP value is infinite at the Leading Edge and Hinge points, and the values printed are the Leading Edge "average value" and the Regular EVD "peak" value underlying the singular Hinge EVD loading, if any. Thus, these points must not be plotted directly by the user. The Jet-Infinity EVD decays slowly from its "peak" value to zero at infinity downstream. The printed values may be used to compute any intermediate loading from the following equations:

Leading Edge Element:

$$\Delta c_{p_{XB}} = \frac{2}{3}\,\Delta c_{p_i}\left[\left(\frac{XB}{\delta_i}\right)^{-\frac{1}{2}} - \left(\frac{XB}{\delta_i}\right)\right] + \Delta c_{p_{i+1}}\left(\frac{XB}{\delta_i}\right)$$

Hinge Element:

$$\Delta c_{p_{XB}} = -\frac{4\beta_i}{\pi}\left[\log(c|XB-XH|) + \frac{\log\left(c\bar{\delta}_{i-1}\right)(XB-XH)}{\bar{\delta}_{i-1}}\right] + \left[\Delta c_{p_i} + \frac{\left(\Delta c_{p_i} - \Delta c_{p_{i-1}}\right)}{\bar{\delta}_{i-1}}(XB-XH)\right]$$

(for XB < XH)

$$\Delta c_{p_{XB}} = -\left[\frac{4\beta_i}{\pi}\,\log(c|XB-XH|) - \frac{\log\left(c\bar{\delta}_i\right)(XB-XH)}{\bar{\delta}_i}\right] + \left[\Delta c_{p_i} - \frac{\left(\Delta c_{p_i} - \Delta c_{p_{i+1}}\right)}{\bar{\delta}_i}(XB-XH)\right]$$

(for XB < XH)

Jet-Infinity Element:

$$\Delta c_{p_{XB}} = \Delta c_{p_i}\, d^2 \left|c\,(XB-1)\right|^{-2}$$

where c is the local chord, $\bar{\delta}$ is the element length, XB is the point where the loading is required, XH is the hinge point, and i refers to the Leading-Edge, Hinge or Jet-Infinity elements, respectively. XB, XH, and $\bar{\delta}$ are normalized by c.

The spanwise loading data are presented in generally the same way as for each fundamental case, except that there is more information. For the linear aerodynamic coefficients (lift, pitching moment, and center of lift), the sectional and total coefficients are given for both zero angle of attack (labels ending with 0) and their variation with angle of attack (labels ending with A). The first two lines of total lift center data are the chordwise locations normalized by reference chord, and in the third and fourth lines they are normalized by the wing semispan.

For the nonlinear induced drag coefficients, three terms are required: zero angle of attack (labels ending with 0), variation with angle of attack squared (labels ending with A2), and the "cross-product" variation with angle of attack (labels ending with X).

A summary table of all total aerodynamic coefficients is printed, with the data taken from the total data of the spanwise loading. Coefficients at zero angle of attack and those which are associated with α and α^2 are shown.

Finally, a summary table of the variations of total lift, pitching moment (about XMC), rolling moment, induced drag (momentum method) and yawing moment about XMC with angle of attack are given. In the sample case, the small negative drag values are not errors, but are simply an indication of inaccuracy resulting from the very coarse spacing used in the example.

It must be noted that in the above description of composite cases, angle of attack equal to zero refers to the basic configuration of the wing as constructed from the scale factors input for all fundamental cases. Since the α-case is a legitimate fundamental case and a multiplication factor may be input for Case 1, it is possible to have a composite case which is labeled "α=0" by the program, but which actually has an arbitrary constant flat plate

loading superimposed on it. The program does not distinguish an input for fundamental Case 1 from the other cases and, therefore, does not "know" that the wing is already at some angle of attack.

This feature may be used with care to obtain the composite loading at some angle of attack of particular interest, say $\alpha=4.3°$, but the user should be aware that he has now forced the loading labels to be misleading. A label of $\alpha=0°$ now refers to the loading at $\alpha=4.3°$, $\alpha=10°$ now means $\alpha=14.3°$, etc. The user should check the scale factor for fundamental Case 1 printed at the top of the page in order to avoid misinterpretation of the composite case output. If it is 0.0, then all the labels are consistent with the printed data.

All of the output data described in Section 5.5 are repeated for each requested composite case in turn.

5.6 Stability Derivative Output

For most runs the program will repeat the above output for all requested composite cases. It will then attempt to read new CMU data and, if no CMU data is found, it will attempt to begin a completely new run. If no new run data is found the job is terminated. For dynamic stability derivative runs, however, the program will automatically begin a second run in which the derivatives due to rolling and yawing are computed. All previously input fundamental cases are altered to represent the effects induced by a yawing rate of one degree/second. Likewise, the extra fundamental case is altered to represent the effects induced by a rolling rate of one degree/second.

The fundamental case output for the second run consists simply of the rolling and yawing derivatives due to yawing (original fundamental cases) and due to rolling (extra fundamental case).

The output for the composite cases again includes the input fundamental case multiplication factors. As with the basic aerodynamic coefficients, the stability derivatives are generally composed of several angle of attack terms, but some of them are also dependent on the rates of yawing or rolling

themselves. The complete formulas for calculation are given in the printout.

Finally, a table of the variation of those derivatives which depend on angle of attack is given. Since only one CMU case can be treated, the program will attempt to begin a completely new run after the second stability derivative run. The termination message is printed, indicating the current run is completed, and if no further input data is given, the job is terminated.

6.0 OPERATIONAL CONSIDERATIONS

6.1 Program Versions Available

The EVD Jet-Wing Computer Program has been written entirely in Fortran IV, and could run on any large computer system. The program was developed on the IBM 360 and 370 systems at Douglas Aircraft Company and converted for use on CDC 6000 series computers. It is expected that only minor changes would be required for running on any other large-scale computing system.

6.2 Overlay Structure

Because of the large size of the program, the overlay feature of FORTRAN is required for operation on most machines. Shown below in Figure 9 is the overlay structure of the Mark II version. This structure requires approximately 141000 octal words of storage on the CDC 6500 and approximately 235000 bytes of storage on the IBM 360 computer. The overlay control cards for CDC 6000 series computers are included in the program listing, Appendix C.

6.3 Peripheral Storage Devices

The program requires the use of six peripheral storage devices, plus any devices required by the local system. These devices may be tape or disk units, and are described as follows:

Unit	Type	Use
1	Binary	Mass storage of downwash influence coefficients and matrices.
2	Binary	Matrix input to solution routine.
3	Binary	Temporary storage during matrix solution.
4	Binary	Temporary storage during matrix solution.
5	BCD	Standard input unit.
6	BCD	Standard printed output unit.

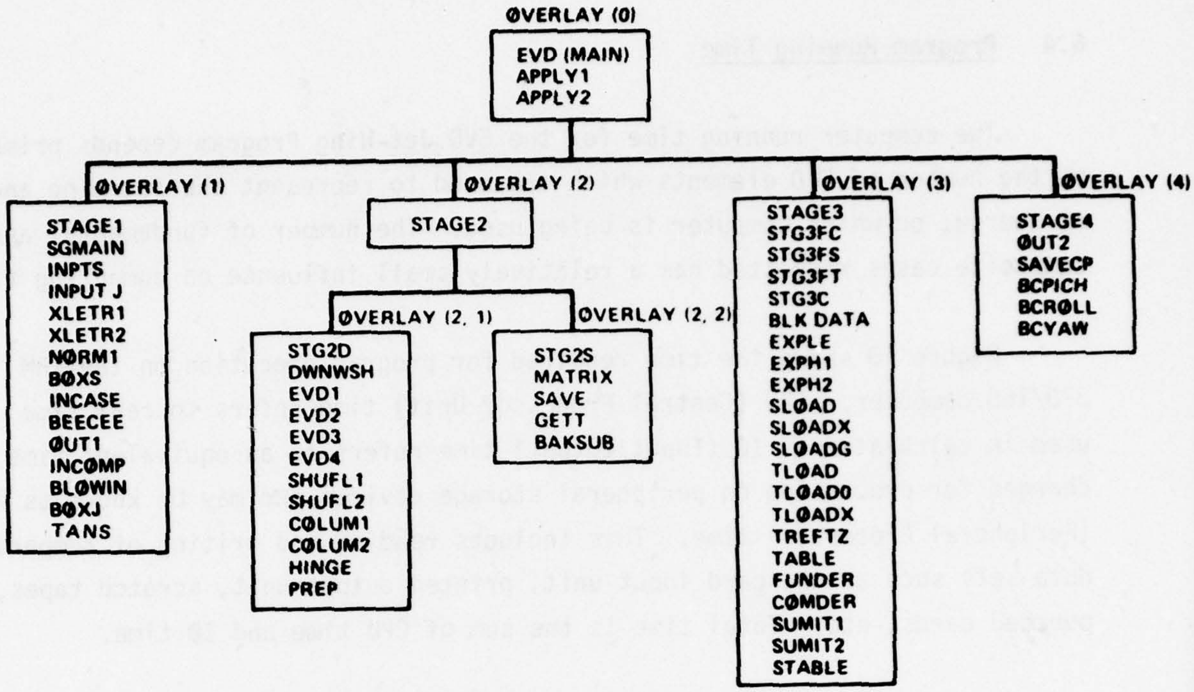


FIGURE 9. PROGRAM OVERLAY STRUCTURE

6.4 Program Running Time

The computer running time for the EVD Jet-Wing Program depends primarily on the number of EVD elements which are used to represent the jet-wing and, of course, on which computer is being used. The number of fundamental and composite cases requested has a relatively small influence on computing time.

Figure 10 shows the time required for program execution on the IBM 370/165 computer. CPU (Central Processor Unit) time refers to real time used in calculation. IØ (Input/Øutput) time refers to an equivalent time charged for processing on peripheral storage devices and may be known as PP (Peripheral Processor) time. This includes reading and writing of temporary data sets such as the card input unit, printed output unit, scratch tapes, punched cards, etc. Total time is the sum of CPU time and IØ time.

Other machines and other installations may require a different amount of time for execution. It is expected that CPU time will vary among machines, but will be representative of the speed of any particular model. On the other hand, IØ time is expected to vary from installation to installation, depending on the facilities available, the demand placed upon them, their cost, etc.

Figure 10 applies to the execution of one value of jet strength (which may be the no-jet case). For each additional jet strength case required in a single run, the total time will be increased by approximately half the total time of the first jet strength case.

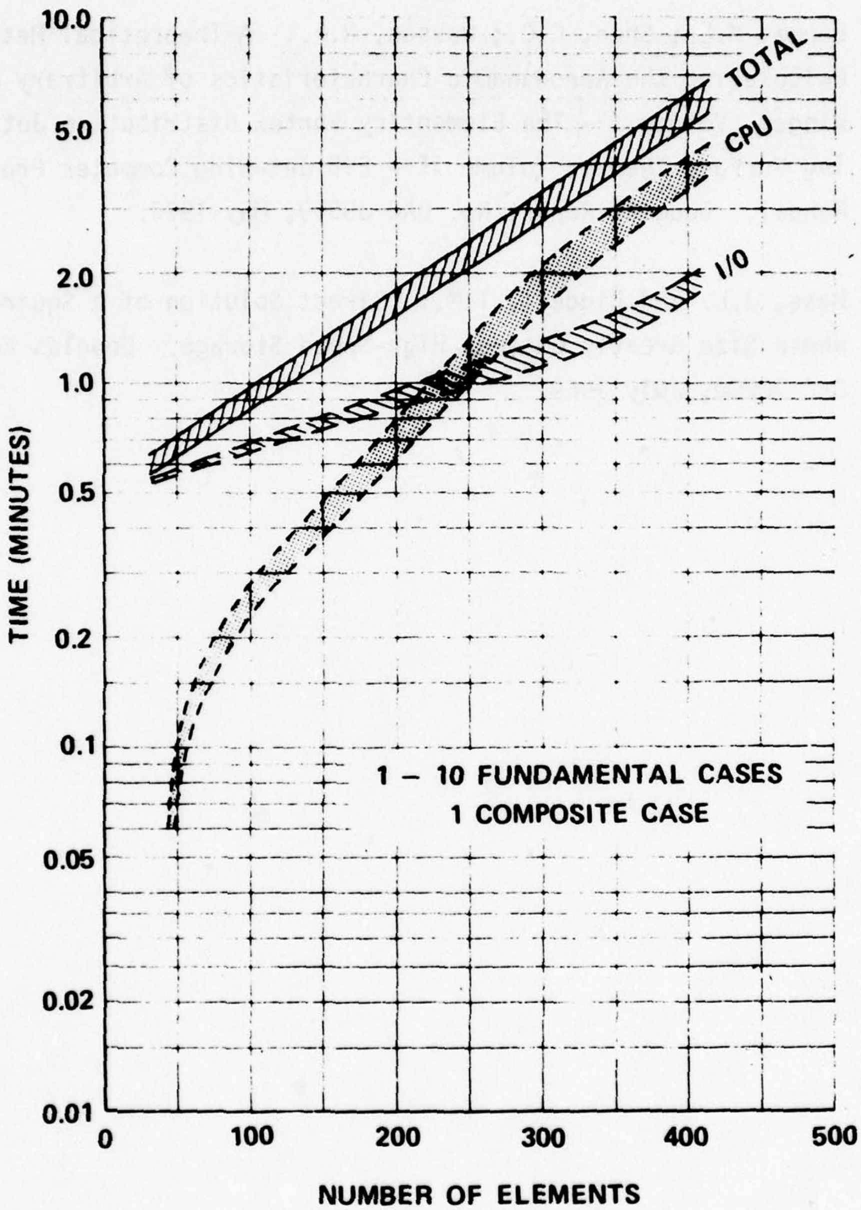


FIGURE 10.

COMPUTER PROGRAM OPERATIONAL TIME REQUIREMENTS

IBM 370/165

REFERENCES

1. Lopez, M.L.; Shen, C.C.; Wasson, N.F.: A Theoretical Method for Calculating the Aerodynamic Characteristics of Arbitrary Jet Flap Wings. Volume I - The Elementary Vortex Distribution Jet-Wing Lifting Surface Theory; Volume II - EVD Jet-Wing Computer Program User's Manual. Douglas Report No. DAC J5519, May 1973.

2. Hess, J.L. and Riddell, T.M.: Direct Solution of a Square Matrix Whose Size Greatly Exceeds High-Speed Storage. Douglas Report No. DAC 70000, July 1969.

APPENDIX A - SAMPLE PROBLEM

Both the input card images and the program printout for a small sample case are given below. It must be noted that this case has been prepared only as a brief example of program input and output. The spacing is very rough, both spanwise and chordwise, and should not be used as a model for preparation of other cases. The fundamental and composite case capabilities of the program are far greater than those demonstrated by the sample case. Because of the crude spacing and few elements, it is expected that the results shown will not accurately predict the wing's aerodynamic characteristics.

INPUT CARD IMAGES

```
 *** ONR SAMPLE CASE ***  RECTANGULAR WING   CMU = 1
                          WITH STABILITY DERIVATIVES
 4.500       4.500       1.000       0.250       0.250
 4 3 0 0 0 2 1 1
 0.9750      0.88750     0.68750     0.2750
 1 1 2 1
 5 6
 0.000       0.100       0.200       0.500       0.900
 0.000       0.100       0.200       0.500       0.800       0.900
 4.500       0.000       1.000
 1 1 1 1
 4
 1.000       1.100       1.500       3.000
0 0 1 0 0
 1.000       1.000       1.000       1.000
 0 0 0 0 1
 0 0 1 0
 0.9000     0 1.000
 1 0.00   2 10.00 3 10.00
9
 1.000       1.000       1.000       1.000
9
```

SAMPLE PROBLEM OUTPUT

```
                    ****************************************
                    *  EVD  JET - WING  COMPUTER  PROGRAM  *
                    ****************************************

     *** ONR SAMPLE CASE ***  RECTANGULAR WING   CMU = 1   WITH STABILITY DERIVATIVES

                          USED           INPUT
                 AREA =   0.888889       4.500000
                 SPAN =   2.000000       4.500000
                 CREF =   0.444444       1.000000
                  XMC =   0.111111       0.250000
                 CMAC =   0.444444       0.999999
               ARATIO =   4.500000       4.500000
                  XCG =   0.111111       0.250000

                NROWS =   4              4
               NCASES =   3              3
                ISYMM =   0              0
               IPRINT =   0              0
               JETFLG =   0              0
               IGTYPE =   2              2
               IHINGE =   0              1

           NUMBER OF WING ELEMENTS =  21
           NUMBER OF JET ELEMENTS  =  16
           TOTAL NUMBER OF ELEMENTS = 37


          ***********************************************************************
          *  ELEMENT GEOMETRY DATA AND FUNDAMENTAL CASE DATA FOR FUNDAMENTAL CASE  1  *
          ***********************************************************************

*** SECTION  1 ***  Y =  0.975000  DELTA =  0.025000  XLEAD =  0.0      XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.0

WING ELEMENTS    NW =  5     TWIST =  0.0            HL =  0.0          THETA S =  0.0
            XB   0.0         0.100000   0.200000   0.500000   0.900000
            XI   0.0         0.044444   0.088889   0.222222   0.400000
           DEL   0.100000    0.100000   0.300000   0.400000   0.100000
           EPS   1.000000    1.000000   1.000000   1.000000   1.000000
          BETA   0.0         0.0        0.0        0.0        0.0
          TYPE   20          10         10         10         10

 JET ELEMENTS    NJ =  4     D =  0.888889    DJ =  0.0           ACTE =  0.0         THETA =  1.000000
            XB   1.000000    1.099999   1.500000   3.000000
            XI   0.444444    0.438889   0.666667   1.333333
           DEL   0.099999    0.400001   1.500000***********
          BETA   0.0         0.0        0.0        0.0
          TYPE   10          10         10         30

*** SECTION  2 ***  Y =  0.887500  DELTA =  0.062500  XLEAD =  0.0      XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.0

WING ELEMENTS    NW =  5     TWIST =  0.0            HL =  0.0          THETA S =  0.0
            XB   0.0         0.100000   0.200000   0.500000   0.900000
            XI   0.0         0.044444   0.088889   0.222222   0.400000
           DEL   0.100000    0.100000   0.300000   0.400000   0.100000
           EPS   1.000000    1.000000   1.000000   1.000000   1.000000
          BETA   0.0         0.0        0.0        0.0        0.0
          TYPE   20          10         10         10         10

 JET ELEMENTS    NJ =  4     D =  0.888889    DJ =  0.0           ACTE =  0.0         THETA =  1.000000
            XB   1.000000    1.099999   1.500000   3.000000
            XI   0.444444    0.438889   0.666667   1.333333
           DEL   0.099999    0.400001   1.500000***********
          BETA   0.0         0.0        0.0        0.0
          TYPE   10          10         10         30

*** SECTION  3 ***  Y =  0.687500  DELTA =  0.137500  XLEAD =  0.0      XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.0

WING ELEMENTS    NW =  6     TWIST =  0.0            HL =  0.0          THETA S =  0.0
            XB   0.0         0.100000   0.200000   0.500000   0.800000   0.900000
            XI   0.0         0.044444   0.088889   0.222222   0.355555   0.400000
           DEL   0.100000    0.100000   0.300000   0.300000   0.100000   0.100000
           EPS   1.000000    1.000000   1.000000   1.000000   1.000000   1.000000
          BETA   0.0         0.0        0.0        0.0        0.0        0.0
          TYPE   20          10         10         10         10         10

 JET ELEMENTS    NJ =  4     D =  0.888889    DJ =  0.0           ACTE =  0.0         THETA =  1.000000
            XB   1.000000    1.099999   1.500000   3.000000
            XI   0.444444    0.488889   0.666667   1.333333
           DEL   0.099999    0.400001   1.500000***********
          BETA   0.0         0.0        0.0        0.0
          TYPE   10          10         10         30


*** SECTION  4 ***  Y =  0.275000  DELTA =  0.275000  XLEAD =  0.0      XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.0

WING ELEMENTS    NW =  5     TWIST =  0.0            HL =  0.0          THETA S =  0.0
            XB   0.0         0.100000   0.200000   0.500000   0.900000
            XI   0.0         0.044444   0.088889   0.222222   0.400000
           DEL   0.100000    0.100000   0.300000   0.400000   0.100000
           EPS   1.000000    1.000000   1.000000   1.000000   1.000000
          BETA   0.0         0.0        0.0        0.0        0.0
          TYPE   20          10         10         10         10

 JET ELEMENTS    NJ =  4     D =  0.388889    DJ =  0.0           ACTE =  0.0         THETA =  1.000000
            XB   1.000000    1.099999   1.500000   3.000000
            XI   0.444444    0.488889   0.666667   1.333333
           DEL   0.099999    0.400001   1.500000***********
          BETA   0.0         0.0        0.0        0.0
          TYPE   10          10         10         30
```

```
*************************************************************************************
*  ELEMENT GEOMETRY DATA AND FUNDAMENTAL CASE DATA FOR FUNDAMENTAL CASE  2  *
*************************************************************************************

*** SECTION  1 ***  Y =  0.975000  DELTA =  0.025000  XLEAD =  0.0       XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.0

WING ELEMENTS    NW =  5      TWIST =  0.0            HL =  0.0           THETA S =  0.0
             XB   0.0          0.100000   0.200000   0.500000   0.900000
            EPS   0.0          0.0        0.0        0.0        0.0
           BETA   0.0          0.0        0.0        0.0        0.0
           TYPE   20           10         10         10         10

 JET ELEMENTS    NJ =  4      D =  0.888889     DJ =  1.000000      ACTE =  0.0         THETA =  1.000000
             XB   1.000000     1.099999   1.500000   3.000000
           BETA   1.000000     0.0        0.0        0.0
           TYPE   43           10         10         30

*** SECTION  2 ***  Y =  0.887500  DELTA =  0.062500  XLEAD =  0.0       XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.0

WING ELEMENTS    NW =  5      TWIST =  0.0            HL =  0.0           THETA S =  0.0
             XB   0.0          0.100000   0.200000   0.500000   0.900000
            EPS   0.0          0.0        0.0        0.0        0.0
           BETA   0.0          0.0        0.0        0.0        0.0
           TYPE   20           10         10         10         10

 JET ELEMENTS    NJ =  4      D =  0.888889     DJ =  1.000000      ACTE =  0.0         THETA =  1.000000
             XB   1.000000     1.099999   1.500000   3.000000
           BETA   1.000000     0.0        0.0        0.0
           TYPE   43           10         10         30

*** SECTION  3 ***  Y =  0.687500  DELTA =  0.137500  XLEAD =  0.0       XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.0

WING ELEMENTS    NW =  6      TWIST =  0.0            HL =  0.0           THETA S =  0.0
             XB   0.0          0.100000   0.200000   0.500000   0.800000   0.900000
            EPS   0.0          0.0        0.0        0.0        0.0        0.0
           BETA   0.0          0.0        0.0        0.0        0.0        0.0
           TYPE   20           10         10         10         10         10

 JET ELEMENTS    NJ =  4      D =  0.888889     DJ =  1.000000      ACTE =  0.0         THETA =  1.000000
             XB   1.000000     1.099999   1.500000   3.000000
           BETA   1.000000     0.0        0.0        0.0
           TYPE   43           10         10         30

*** SECTION  4 ***  Y =  0.275000  DELTA =  0.275000  XLEAD =  0.0       XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.0

WING ELEMENTS    NW =  5      TWIST =  0.0            HL =  0.0           THETA S =  0.0
             XB   0.0          0.100000   0.200000   0.500000   0.900000
            EPS   0.0          0.0        0.0        0.0        0.0
           BETA   0.0          0.0        0.0        0.0        0.0
           TYPE   20           10         10         10         10

 JET ELEMENTS    NJ =  4      D =  0.888889     DJ =  1.000000      ACTE =  0.0         THETA =  1.000000
             XB   1.000000     1.099999   1.500000   3.000000
           BETA   1.000000     0.0        0.0        0.0
           TYPE   43           10         10         30


*************************************************************************************
*  ELEMENT GEOMETRY DATA AND FUNDAMENTAL CASE DATA FOR FUNDAMENTAL CASE  3  *
*************************************************************************************

*** SECTION  1 ***  Y =  0.975000  DELTA =  0.025000  XLEAD =  0.0       XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.0

WING ELEMENTS    NW =  5      TWIST =  0.0            HL =  0.0           THETA S =  0.0
             XB   0.0          0.100000   0.200000   0.500000   0.900000
            EPS   0.0          0.0        0.0        0.0        0.0
           BETA   0.0          0.0        0.0        0.0        0.0
           TYPE   20           10         10         10         10

 JET ELEMENTS    NJ =  4      D =  0.888889     DJ =  0.0           ACTE =  0.0         THETA =  0.0
             XB   1.000000     1.099999   1.500000   3.000000
           BETA   0.0          0.0        0.0        0.0
           TYPE   43           10         10         30

*** SECTION  2 ***  Y =  0.887500  DELTA =  0.062500  XLEAD =  0.0       XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.0

WING ELEMENTS    NW =  5      TWIST =  0.0            HL =  0.0           THETA S =  0.0
             XB   0.0          0.100000   0.200000   0.500000   0.900000
            EPS   0.0          0.0        0.0        0.0        0.0
           BETA   0.0          0.0        0.0        0.0        0.0
           TYPE   20           10         10         10         10

 JET ELEMENTS    NJ =  4      D =  0.888889     DJ =  0.0           ACTE =  0.0         THETA =  0.0
             XB   1.000000     1.099999   1.500000   3.000000
           BETA   0.0          0.0        0.0        0.0
           TYPE   43           10         10         30

*** SECTION  3 ***  Y =  0.687500  DELTA =  0.137500  XLEAD =  0.0       XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.0

WING ELEMENTS    NW =  6      TWIST =  0.0            HL =  0.0           THETA S =  0.0
             XB   0.0          0.100000   0.200000   0.500000   0.800000   0.900000
            EPS   0.0          0.0        0.0        0.0        0.0        1.000000
           BETA   0.0          0.0        0.0        0.0        0.0        1.000000
           TYPE   20           10         10         10         10         42

 JET ELEMENTS    NJ =  4      D =  0.888889     DJ =  0.0           ACTE =  0.0         THETA =  1.000000
             XB   1.000000     1.099999   1.500000   3.000000
           BETA   0.0          0.0        0.0        0.0
           TYPE   43           10         10         30

*** SECTION  4 ***  Y =  0.275000  DELTA =  0.275000  XLEAD =  0.0       XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.0

WING ELEMENTS    NW =  5      TWIST =  0.0            HL =  0.0           THETA S =  0.0
             XB   0.0          0.100000   0.200000   0.500000   0.900000
            EPS   0.0          0.0        0.0        0.0        0.0
           BETA   0.0          0.0        0.0        0.0        0.0
           TYPE   20           10         10         10         10

 JET ELEMENTS    NJ =  4      D =  0.888889     DJ =  0.0           ACTE =  0.0         THETA =  0.0
             XB   1.000000     1.099999   1.500000   3.000000
           BETA   0.0          0.0        0.0        0.0
           TYPE   43           10         10         30
```

```
*******************************************
*  SECTIONAL JET BLOWING COEFFICIENTS    *
*******************************************

                 ROW      CMU
                  1     1.000000
                  2     1.000000
                  3     1.000000
                  4     1.000000
```

```
**********************************************************************
*  ELEMENT GEOMETRY DATA AND FUNDAMENTAL CASE DATA FOR FUNDAMENTAL CASE  4  *
**********************************************************************

*** SECTION  1 ***  Y =  0.975000  DELTA =  0.025000  XLEAD =  0.0       XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.0

WING ELEMENTS    NW =  5     TWIST =  0.0            HL =  0.0            THETA S =  0.0
            XB   0.0          0.100000   0.200000   0.500000   0.900000
           EPS  -0.400000    -0.200000   0.199999   0.900000   1.400000
          BETA   0.0          0.0        0.0        0.0        0.0
          TYPE   20           10         10         10         10

 JET ELEMENTS    NJ =  4     D =  0.888889     DJ =  0.0            ACTE =  0.0          THETA =  1.500000
            XB   1.000000     1.099999   1.500000   3.000000
          BETA   0.0          0.0        0.0        0.0
          TYPE   43           10         10         30

*** SECTION  2 ***  Y =  0.887500  DELTA =  0.062500  XLEAD =  0.0       XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.0

WING ELEMENTS    NW =  5     TWIST =  0.0            HL =  0.0            THETA S =  0.0
            XB   0.0          0.100000   0.200000   0.500000   0.900000
           EPS  -0.400000    -0.200000   0.199999   0.900000   1.400000
          BETA   0.0          0.0        0.0        0.0        0.0
          TYPE   20           10         10         10         10

 JET ELEMENTS    NJ =  4     D =  0.888889     DJ =  0.0            ACTE =  0.0          THETA =  1.500000
            XB   1.000000     1.099999   1.500000   3.000000
          BETA   0.0          0.0        0.0        0.0
          TYPE   43           10         10         30

*** SECTION  3 ***  Y =  0.687500  DELTA =  0.137500  XLEAD =  0.0       XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.0

WING ELEMENTS    NW =  6     TWIST =  0.0            HL =  0.0            THETA S =  0.0
            XB   0.0          0.100000   0.200000   0.500000   0.800000   0.900000
           EPS  -0.400000    -0.200000   0.199999   0.800000   1.199999   1.400000
          BETA   0.0          0.0        0.0        0.0        0.0        0.0
          TYPE   20           10         10         10         10         42

 JET ELEMENTS    NJ =  4     D =  0.888889     DJ =  0.0            ACTE =  0.0          THETA =  1.500000
            XB   1.000000     1.099999   1.500000   3.000000
          BETA   0.0          0.0        0.0        0.0
          TYPE   43           10         10         30

*** SECTION  4 ***  Y =  0.275000  DELTA =  0.275000  XLEAD =  0.0       XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.0

WING ELEMENTS    NW =  5     TWIST =  0.0            HL =  0.0            THETA S =  0.0
            XB   0.0          0.100000   0.200000   0.500000   0.900000
           EPS  -0.400000    -0.200000   0.199999   0.900000   1.400000
          BETA   0.0          0.0        0.0        0.0        0.0
          TYPE   20           10         10         10         10

 JET ELEMENTS    NJ =  4     D =  0.888889     DJ =  0.0            ACTE =  0.0          THETA =  1.500000
            XB   1.000000     1.099999   1.500000   3.000000
          BETA   0.0          0.0        0.0        0.0
          TYPE   43           10         10         30
```

* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *

SECTION 1 Y = 0.975000 CHORD = 0.444444

	I	XB	CASE 1	CASE 2	CASE 3	CASE 4	CASE 5	CASE 6	CASE 7	CASE 8	CASE 9	CASE 10
WING	1	0.0	0.129570	0.015231	0.006761	0.028993	0.0	0.0	0.0	0.0	0.0	0.0
	2	0.100000	0.062480	0.009628	0.004529	0.041840	0.0	0.0	0.0	0.0	0.0	0.0
	3	0.200000	0.036112	0.007270	0.003624	0.042261	0.0	0.0	0.0	0.0	0.0	0.0
	4	0.500000	0.016367	0.006730	0.004280	0.044040	0.0	0.0	0.0	0.0	0.0	0.0
	5	0.900000	0.007476	0.016049	0.005106	0.033779	0.0	0.0	0.0	0.0	0.0	0.0
JET												
	22	1.000000	0.005186	0.058976	0.003649	0.019742	0.0	0.0	0.0	0.0	0.0	0.0
	23	1.099999	0.003300	0.009183	0.002402	0.008359	0.0	0.0	0.0	0.0	0.0	0.0
	24	1.500000	0.001203	0.001247	0.000598	0.002077	0.0	0.0	0.0	0.0	0.0	0.0
	25	3.000000	0.000202	0.000111	0.000026	0.000261	0.0	0.0	0.0	0.0	0.0	0.0

DETAILED LEADING EDGE LOADING

1	0.020000	0.168371	0.022600	0.010083	0.047722
2	0.040000	0.127018	0.015845	0.007136	0.039566
3	0.060000	0.097176	0.012793	0.005832	0.038460
4	0.080000	0.077456	0.010931	0.005057	0.039619
5	0.100000	0.062480	0.009628	0.004529	0.041840

* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *

SECTION 2 Y = 0.887500 CHORD = 0.444444

	I	XB	CASE 1	CASE 2	CASE 3	CASE 4	CASE 5	CASE 6	CASE 7	CASE 8	CASE 9	CASE 10
WING	6	0.0	0.185800	0.024182	0.010750	0.056734	0.0	0.0	0.0	0.0	0.0	0.0
	7	0.100000	0.107434	0.017094	0.008019	0.068596	0.0	0.0	0.0	0.0	0.0	0.0
	8	0.200000	0.064708	0.013036	0.006499	0.067874	0.0	0.0	0.0	0.0	0.0	0.0
	9	0.500000	0.029236	0.012062	0.007998	0.070135	0.0	0.0	0.0	0.0	0.0	0.0
	10	0.900000	0.012979	0.027342	0.011413	0.051704	0.0	0.0	0.0	0.0	0.0	0.0
JET												
	26	1.000000	0.008878	0.071214	0.008014	0.030824	0.0	0.0	0.0	0.0	0.0	0.0
	27	1.099999	0.005497	0.015626	0.004878	0.014111	0.0	0.0	0.0	0.0	0.0	0.0
	28	1.500000	0.001876	0.001991	0.000959	0.003273	0.0	0.0	0.0	0.0	0.0	0.0
	29	3.000000	0.000287	0.000155	0.000032	0.000369	0.0	0.0	0.0	0.0	0.0	0.0

DETAILED LEADING EDGE LOADING

1	0.020000	0.273687	0.036243	0.016196	0.090728
2	0.040000	0.189277	0.025879	0.011673	0.072112
3	0.060000	0.150051	0.021396	0.009764	0.067293
4	0.080000	0.125341	0.018803	0.008694	0.066906
5	0.100000	0.107434	0.017094	0.008019	0.068596

* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *

SECTION 3 Y = 0.687500 CHORD = 0.444444

	I	XB	CASE 1	CASE 2	CASE 3	CASE 4	CASE 5	CASE 6	CASE 7	CASE 8	CASE 9	CASE 10
WING	11	0.0	0.230761	0.034686	0.014448	0.091838	0.0	0.0	0.0	0.0	0.0	0.0
	12	0.100000	0.140898	0.025102	0.011127	0.095908	0.0	0.0	0.0	0.0	0.0	0.0
	13	0.200000	0.089584	0.019423	0.009298	0.090418	0.0	0.0	0.0	0.0	0.0	0.0
	14	0.500000	0.044098	0.018778	0.012470	0.089254	0.0	0.0	0.0	0.0	0.0	0.0
	15	0.800000	0.025147	0.027830	0.032386	0.073930	0.0	0.0	0.0	0.0	0.0	0.0
	16	0.900000	0.020025	0.035635	0.085626	0.062237	0.0	0.0	0.0	0.0	0.0	0.0
JET												
	30	1.000000	0.013414	0.076381	0.026293	0.039217	0.0	0.0	0.0	0.0	0.0	0.0
	31	1.099999	0.008238	0.019171	0.011354	0.019388	0.0	0.0	0.0	0.0	0.0	0.0
	32	1.500000	0.002674	0.002923	0.001175	0.004705	0.0	0.0	0.0	0.0	0.0	0.0
	33	3.000000	0.000348	0.000179	0.000040	0.000442	0.0	0.0	0.0	0.0	0.0	0.0

DETAILED LEADING EDGE LOADING

1	0.020000	0.341409	0.052102	0.021837	0.143841
2	0.040000	0.238066	0.037353	0.015827	0.110679
3	0.060000	0.190841	0.031040	0.013332	0.099851
4	0.080000	0.161645	0.027436	0.011965	0.096198
5	0.100000	0.140898	0.025102	0.011127	0.095908

* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *

SECTION 4 Y = 0.275000 CHORD = 0.444444

	I	XB	CASE 1	CASE 2	CASE 3	CASE 4	CASE 5	CASE 6	CASE 7	CASE 8	CASE 9	CASE 10
WING	17	0.0	0.261299	0.044340	0.011430	0.121582	0.0	0.0	0.0	0.0	0.0	0.0
	18	0.100000	0.152285	0.032007	0.007930	0.117146	0.0	0.0	0.0	0.0	0.0	0.0
	19	0.200000	0.105206	0.024658	0.005608	0.106311	0.0	0.0	0.0	0.0	0.0	0.0
	20	0.500000	0.054223	0.022370	0.004253	0.101691	0.0	0.0	0.0	0.0	0.0	0.0
	21	0.900000	0.025307	0.036868	0.002913	0.071827	0.0	0.0	0.0	0.0	0.0	0.0
JET												
	34	1.000000	0.017243	0.078280	0.002095	0.044616	0.0	0.0	0.0	0.0	0.0	0.0
	35	1.099999	0.010665	0.020564	0.001440	0.022860	0.0	0.0	0.0	0.0	0.0	0.0
	36	1.500000	0.003524	0.003578	0.000644	0.005991	0.0	0.0	0.0	0.0	0.0	0.0
	37	3.000000	0.000403	0.000220	0.000081	0.000522	0.0	0.0	0.0	0.0	0.0	0.0

DETAILED LEADING EDGE LOADING

1	0.020000	0.387136	0.066587	0.017101	0.188462
2	0.040000	0.270667	0.047717	0.012172	0.142595
3	0.060000	0.217741	0.039630	0.010023	0.126296
4	0.080000	0.185229	0.035007	0.008767	0.119495
5	0.100000	0.152285	0.032007	0.007930	0.117146

SPANWISE LOADING FOR FUNDAMENTAL CASE 1

SECTION	Y	CLG	CLMU	CL	CDG	CDMU	CS	CD	CMU	GAMMA	ALFIN
		LIFT			INDUCED DRAG						
1	0.975000	0.034284	0.017453	0.051737	0.0005984	0.0001523	0.0002930	0.0004577	1.0000000	0.0081921	0.0427911
2	0.887500	0.056186	0.017453	0.073639	0.0009806	0.0001523	0.0006025	0.0005304	1.0000000	0.0133976	0.0218746
3	0.687500	0.076014	0.017453	0.093468	0.0013267	0.0001523	0.0009294	0.0005496	1.0000000	0.0181988	0.0120300
4	0.275000	0.089566	0.017453	0.107020	0.0015632	0.0001523	0.0011917	0.0005239	1.0000000	0.0215896	0.0065474
TOTAL		0.078903	0.017453	0.096356	0.0013771	0.0001523	0.0010010	0.0005285	0.9999996		0.0005033

SECTION	Y	CMG	CMMU	CMT	CM		XCP/C	XCL/C	
		PITCHING MOMENT					LIFT CENTER		
1	0.975000	-0.007645	-0.017453	0.017453	-0.007645	* *	0.222985	0.485106	
2	0.887500	-0.013398	-0.017453	0.017453	-0.013388	* *	0.238284	0.418818	
3	0.687500	-0.019396	-0.017453	0.017453	-0.019396	* *	0.255167	0.394250	
4	0.275000	-0.023761	-0.017453	0.017453	-0.023761	* *	0.265285	0.385106	
TOTAL		-0.020458	-0.017453	0.017453	-0.020458	(APEX)	0.259282	0.393450	(X/CREF)
		-0.000732	-0.013090	0.013090	-0.000732	(XMC)	0.115236	0.174867	(X/B/2)

SPANWISE LOADING FOR FUNDAMENTAL CASE 2

SECTION	Y	CLG	CLMU	CL	CDG	CDMU	CS	CD	CMU	GAMMA	ALFIN
		LIFT			INDUCED DRAG						
1	0.975000	0.013256	0.017453	0.030710	0.0	0.0001523	0.0000040	0.0001483	1.0000000	0.0044180	0.0254144
2	0.887500	0.021353	0.017453	0.038806	0.0	0.0001523	0.0000102	0.0001421	1.0000000	0.0068849	0.0122251
3	0.687500	0.028446	0.017453	0.045899	0.0	0.0001523	0.0000210	0.0001313	1.0000000	0.0089218	0.0057449
4	0.275000	0.033527	0.017453	0.050980	0.0	0.0001523	0.0000343	0.0001180	1.0000000	0.0103034	0.0022070
TOTAL		0.029594	0.017453	0.047047	0.0	0.0001523	0.0000261	0.0001262	0.9999996		0.0001164

SECTION	Y	CMG	CMMU	CMT	CM		XCP/C	XCL/C	
		PITCHING MOMENT					LIFT CENTER		
1	0.975000	-0.007630	-0.017453	0.0	-0.025283	* *	0.590655	0.823298	
2	0.887500	-0.012071	-0.017453	0.0	-0.029524	* *	0.565305	0.760812	
3	0.687500	-0.015144	-0.017453	0.0	-0.032597	* *	0.532393	0.710203	
4	0.275000	-0.017045	-0.017453	0.0	-0.034498	* *	0.508389	0.676695	
TOTAL		-0.015440	-0.017453	0.0	-0.032893	(APEX)	0.521710	0.699142	(X/CREF)
		-0.008041	-0.013090	0.0	-0.021131	(XMC)	0.231871	0.310730	(X/B/2)

SPANWISE LOADING FOR FUNDAMENTAL CASE 3

SECTION	Y	CLG	CLMU	CL	CDG	CDMU	CS	CD	CMU	GAMMA	ALFIN
		LIFT			INDUCED DRAG						
1	0.975000	0.004811	0.0	0.004811	0.0	0.0	0.0000008	-0.0000008	1.0000000	0.0013792	0.0050237
2	0.887500	0.009230	0.0	0.009230	0.0	0.0	0.0000020	-0.0000020	1.0000000	0.0026257	0.0014837
3	0.687500	0.024510	0.017453	0.041964	0.0000977	0.0001523	0.0000036	0.0002463	1.0000000	0.0066333	0.0108795
4	0.275000	0.005419	0.0	0.005419	0.0	0.0	0.0000023	-0.0000023	1.0000000	0.0014751	0.0095400
TOTAL		0.011115	0.004800	0.015915	0.0000269	0.0000419	0.0000025	0.0000662	0.9999996		0.0000785

SECTION	Y	CMG	CMMU	CMT	CM		XCP/C	XCL/C	
		PITCHING MOMENT					LIFT CENTER		
1	0.975000	-0.002248	0.0	0.0	-0.002248	* *	0.467185	0.467185	
2	0.687500	-0.004608	0.0	0.0	-0.004608	* *	0.499208	0.499208	
3	0.687500	-0.016228	-0.017453	0.001745	-0.031936	* *	0.662069	0.802619	
4	0.275000	-0.001887	0.0	0.0	-0.001887	* *	0.348260	0.348260	
TOTAL		-0.006189	-0.004800	0.000480	-0.010509	(APEX)	0.556798	0.690460	(X/CREF)
		-0.003410	-0.003600	0.000480	-0.006530	(XMC)	0.247466	0.306871	(X/B/2)

SPANWISE LOADING FOR FUNDAMENTAL CASE 4

SECTION	Y	CLG	CLMU	CL	CDG	CDMU	CS	CD	CMU	GAMMA	ALFIN
		LIFT			INDUCED DRAG						
1	0.975000	0.040381	0.026180	0.066561	0.0003055	0.0003427	0.0000147	0.0006335	1.0000000	0.0101972	0.0550330
2	0.887500	0.065122	0.026180	0.091302	0.0004685	0.0003427	0.0000552	0.0007550	1.0000000	0.0164325	0.0277257
3	0.687500	0.086605	0.026180	0.112785	0.0005723	0.0003427	0.0001472	0.0007678	1.0000000	0.0219236	0.0144727
4	0.275000	0.100914	0.026180	0.127094	0.0006315	0.0003427	0.0002580	0.0007162	1.0000000	0.0256589	0.0069844
TOTAL		0.089478	0.026180	0.115658	0.0005785	0.0003427	0.0001901	0.0007311	0.9999996		0.0007183

SECTION	Y	CMG	CMMU	CMT	CM		XCP/C	XCL/C	
		PITCHING MOMENT					LIFT CENTER		
1	0.975000	-0.018607	-0.026180	0.0	-0.044787	* *	0.460776	0.672865	
2	0.887500	-0.029265	-0.026180	0.0	-0.055445	* *	0.449387	0.607270	
3	0.687500	-0.037602	-0.026180	0.0	-0.063782	* *	0.434182	0.565522	
4	0.275000	-0.042490	-0.026180	0.0	-0.068670	* *	0.421055	0.540310	
TOTAL		-0.038299	-0.026180	0.0	-0.064479	(APEX)	0.428023	0.557493	(X/CREF)
		-0.015929	-0.019635	0.0	-0.035564	(XMC)	0.190232	0.247775	(X/B/2)

LIFT COEFFICIENT DERIVATIVE DUE TO PITCHING ABOUT XCG, CLQ = 0.089478
PITCHING MOMENT COEFFICIENT DERIVATIVE ABOUT ORIGIN DUE TO PITCHING ABOUT XCG, CMQ = -0.038299
PITCHING MOMENT COEFF DERIVATIVE ABOUT XMC DUE TO PITCHING ABOUT XCG, CMQMC = -0.015929

****

* TOTAL AERODYNAMIC COEFFICIENTS *

****

	CASE 1	CASE 2	CASE 3	CASE 4	CASE 5	CASE 6	CASE 7	CASE 8	CASE 9	CASE 10
CCLG	0.0785028	0.0295941	0.0111152	0.0894784	0.0	0.0	0.0	0.0	0.0	0.0
CCLJ	0.0174533	0.0174533	0.0047997	0.0261799	0.0	0.0	0.0	0.0	0.0	0.0
** CCL	0.0963561	0.0470474	0.0159149	0.1156583	0.0	0.0	0.0	0.0	0.0	0.0
CCDG	0.0013771	0.0	0.0000269	0.0005785	0.0	0.0	0.0	0.0	0.0	0.0
CCDJ	0.0001523	0.0001523	0.0000419	0.0003427	0.0	0.0	0.0	0.0	0.0	0.0
CCS	0.0010010	0.0000261	0.0000025	0.0001901	0.0	0.0	0.0	0.0	0.0	0.0
CCD	0.0005285	0.0001262	0.0000662	0.0007311	0.0	0.0	0.0	0.0	0.0	0.0
** CDITZ	0.0005033	0.0001164	0.0000785	0.0007183	0.0	0.0	0.0	0.0	0.0	0.0
** CCJ	0.9999996	0.9999996	0.9999996	0.9999996	0.0	0.0	0.0	0.0	0.0	0.0
CCMG	-0.0204581	-0.0154396	-0.0061899	-0.0362988	0.0	0.0	0.0	0.0	0.0	0.0
CCMJ	-0.0174533	-0.0174533	-0.0047997	-0.0261799	0.0	0.0	0.0	0.0	0.0	0.0
CCMT	0.0174533	0.0	0.0004800	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CCM	-0.0204581	-0.0328929	-0.0105086	-0.0644787	0.0	0.0	0.0	0.0	0.0	0.0
CXCP	0.2592817	0.5217092	0.5567981	0.4280229	0.0	0.0	0.0	0.0	0.0	0.0
CXCL	0.3934504	0.6991421	0.6904603	0.5574928	0.0	0.0	0.0	0.0	0.0	0.0
CXCPB	0.1152363	0.2318709	0.2474658	0.1902324	0.0	0.0	0.0	0.0	0.0	0.0
CXCLB	0.1748668	0.3107298	0.3068712	0.2477745	0.0	0.0	0.0	0.0	0.0	0.0
CCMGMC	-0.0007324	-0.0080410	-0.0034101	-0.0159292	0.0	0.0	0.0	0.0	0.0	0.0
CCMJMC	-0.0130900	-0.0130900	-0.0035997	-0.0196349	0.0	0.0	0.0	0.0	0.0	0.0
CCMTMC	0.0130900	0.0	0.0004800	0.0	0.0	0.0	0.0	0.0	0.0	0.0
** CCMMC	-0.0007324	-0.0211310	-0.0065299	-0.0355642	0.0	0.0	0.0	0.0	0.0	0.0
CLLG	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CLLJ	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
* CLL	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CNJ	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
* CNIMC	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
* CCY	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CBGR	0.0358229	0.0134640	0.0067121	0.0409301	0.0	0.0	0.0	0.0	0.0	0.0
CBGL	0.0358229	0.0134640	0.0067121	0.0403301	0.0	0.0	0.0	0.0	0.0	0.0
CBJR	0.0087266	0.0087266	0.0032998	0.0130900	0.0	0.0	0.0	0.0	0.0	0.0
CBJL	0.0087266	0.0087266	0.0032998	0.0130900	0.0	0.0	0.0	0.0	0.0	0.0
CBR	0.0445495	0.0221906	0.0100119	0.0539200	0.0	0.0	0.0	0.0	0.0	0.0
CBL	0.0445495	0.0221906	0.0100119	0.0539200	0.0	0.0	0.0	0.0	0.0	0.0
CPMBR	0.4623423	0.4716647	0.6290900	0.4662008	0.0	0.0	0.0	0.0	0.0	0.0
CPMBL	0.4623423	0.4716647	0.6290900	0.4662008	0.0	0.0	0.0	0.0	0.0	0.0

```
*************************************************
*  CHORDWISE LOADING FOR COMPOSITE CASE  1  *
*************************************************
```

FUNDAMENTAL CASE FACTORS

A(1)	A(2)	A(3)	A(4)	A(5)	A(6)	A(7)	A(8)	A(9)	A(10)
0.0	10.000000	10.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0

*** NOTE *** EACH LEADING EDGE CP VALUE IS THE AVERAGE VALUE OF THE SINGULAR DISTRIBUTION
DO NOT PLOT THESE LOADING POINTS DIRECTLY

SECTION 1 Y = 0.975000 CHORD = 0.444444

WING						
XB	0.0	0.100000	0.200000	0.500000	0.900000	
CP(A=0)	0.219927	0.141564	0.108941	0.110103	0.211553	
CP(A=1)	0.129570	0.062480	0.036112	0.016367	0.007476	
JET						
XB	1.000000	1.099999	1.500000	3.000000		
CP(A=0)	0.626244	0.115851	0.018450	0.001371		
CP(A=1)	0.005186	0.003300	0.001203	0.000202		

SECTION 2 Y = 0.887500 CHORD = 0.444444

WING						
XB	0.0	0.100000	0.200000	0.500000	0.900000	
CP(A=0)	0.349325	0.251132	0.195345	0.200600	0.387549	
CP(A=1)	0.185800	0.107434	0.064708	0.029236	0.012979	
JET						
XB	1.000000	1.099999	1.500000	3.000000		
CP(A=0)	0.792272	0.205033	0.029491	0.001867		
CP(A=1)	0.008878	0.005457	0.001876	0.000287		

SECTION 3 Y = 0.687500 CHORD = 0.444444

WING						
XB	0.0	0.100000	0.200000	0.500000	0.800000	0.900000
CP(A=0)	0.491340	0.362289	0.287105	0.312486	0.602161	1.212608
CP(A=1)	0.230761	0.140898	0.089584	0.044098	0.025147	0.020025
JET						
XB	1.000000	1.099999	1.500000	3.000000		
CP(A=0)	1.026743	0.305252	0.040979	0.002192		
CP(A=1)	0.013414	0.008238	0.002674	0.000348		

SECTION 4 Y = 0.275000 CHORD = 0.444444

WING						
XB	0.0	0.100000	0.200000	0.500000	0.900000	
CP(A=0)	0.557696	0.399370	0.304666	0.266234	0.397805	
CP(A=1)	0.261299	0.162285	0.105206	0.054223	0.025307	
JET						
XB	1.000000	1.099999	1.500000	3.000000		
CP(A=0)	0.303744	0.220041	0.042222	0.003008		
CP(A=1)	0.017243	0.010685	0.003524	0.000403		

```
**************************
*  COMPOSITE  CASE  1  *
**************************
```

FUNDAMENTAL CASE FACTORS

A(1)	A(2)	A(3)	A(4)	A(5)	A(6)	A(7)	A(8)	A(9)	A(10)
0.0	10.000000	10.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0

SECTION	Y	...... LIFT ...... CLGO / CLGA	CLMUO / CLMUA	CLO / CLA	...... PITCHING MOMENT ...... CMGO / CMGA	CMMUO / CMMUA	CMTO / CMTA	CMO / CMA		... LIFT CENTER ... XCPO/C / XCPA/C	XCLO/C / XCLA/C
1	0.975000	0.180674	0.174533	0.355206	-0.100776	-0.174533	0.0	-0.275309		0.557779	0.775067
		0.034284	0.017453	0.051737	-0.007645	-0.017453	0.017453	-0.007645		0.222985	0.485106
2	0.887500	0.305825	0.174533	0.480358	-0.166784	-0.174533	0.0	-0.341317		0.545357	0.710547
		0.056186	0.017453	0.073639	-0.013388	-0.017453	0.017453	-0.013388		0.238284	0.418818
3	0.687500	0.529560	0.349066	0.878626	-0.313718	-0.349066	0.017453	-0.645331		0.592413	0.754342
		0.076014	0.017453	0.093468	-0.019396	-0.017453	0.017453	-0.019396		0.255167	0.394250
4	0.275000	0.389460	0.174533	0.563993	-0.189319	-0.174533	0.0	-0.363852		0.486108	0.645136
		0.089566	0.017453	0.107020	-0.023761	-0.017453	0.017453	-0.023761		0.265285	0.385106
TOTAL		0.407093	0.222529	0.629623	-0.216285	-0.222529	0.004800	-0.434014	(APEX)	0.531290	0.696948
		0.078903	0.017453	0.096356	-0.020458	-0.017453	0.017453	-0.020458	(APEX)	0.259282	0.393450
					-0.114511	-0.166897	0.004800	-0.276609	(XMC)	0.236129	0.309754
					-0.000732	-0.013090	0.013090	-0.000732	(XMC)	0.115236	0.174867

SECTION	Y	...... INDUCED DRAG ...... CDGO / CDGX / CDGA2	CDMUO / CDMUX / CDMUA2	CSO / CSX / CSA2	CDO / CDX / CDA2	GAMMAO / GAMMA2	ALFINO / ALFINA	CTO	CMU
1	0.975000	0.0	0.0152309	0.0008442	0.0143867	0.0579720	0.3043813	0.9856133	1.0000000
		0.0031533	0.0030462	0.0009947	0.0052048				
		0.0005984	0.0001523	0.0002930	0.0004577	0.0081921	0.0427911		
2	0.887500	0.0	0.0152309	0.0021298	0.0131011	0.0951066	0.1370882	0.9868990	1.0000000
		0.0053377	0.0030462	0.0022656	0.0061182				
		0.0009306	0.0001523	0.0006025	0.0005304	0.0133976	0.0218746		
3	0.687500	0.0195420	0.0609235	0.0042135	0.0762519	0.1555502	0.1662443	0.9237481	1.0000000
		0.0095344	0.0060923	0.0039578	0.0116690				
		0.0013267	0.0001523	0.0009294	0.0005496	0.0181988	0.0120300		
4	0.275000	0.0	0.0152309	0.0054284	0.0098025	0.1177849	0.1174700	0.9901975	1.0000000
		0.0067974	0.0030462	0.0050868	0.0047568				
		0.0015632	0.0001523	0.0011917	0.0005239	0.0215896	0.0065474		
TOTAL		0.0053741	0.0277963	0.0044528	0.0287176		0.0451221	0.9712820	0.9999996
		0.0071854	0.0038839	0.0042190	0.0068502		0.0101239		
		0.0013771	0.0001523	0.0010010	0.0005285		0.0005033		

*** TOTAL AERODYNAMIC COEFFICIENTS ***

		ALPHA=0	ALPHA	ALPHA**2
	CCLG	0.407093	0.078903	
	CCLJ	0.222529	0.017453	
**	CCL	0.629623	0.096356	
	CCDG	0.0053741	0.0071854	0.0013771
	CCDJ	0.0277963	0.0038839	0.0001523
	CCS	0.0044526	0.0042190	0.0010010
	CCD	0.0287176	0.0068502	0.0005285
**	CDITZ	0.0451221	0.0101239	0.0005033
**	CCJ	0.9999996		
	CCMG	-0.216285	-0.020458	
	CCMJ	-0.222529	-0.017453	
	CCMT	0.004800	0.017453	
	CCM	-0.434014	-0.020458	
	CXCP	0.531290	0.259282	
	CXCL	0.696948	0.393450	
	CXCPB	0.236129	0.115236	
	CXCLB	0.309754	0.174867	
	CCMGMC	-0.114511	-0.000732	
	CCMJMC	-0.166897	-0.013090	
	CCMTMC	0.004800	0.013090	
**	CCMMC	-0.276609	-0.000732	
	CLLG	0.0	0.0	
	CLLJ	0.0	0.0	
*	CLL	0.0	0.0	
	CNJ	0.0		
*	CNIMC	0.0	0.0	0.0
*	CCY	0.0	0.0	0.0
	CBGR	0.201761	0.035823	
	CBGL	0.201761	0.035823	
	CBJR	0.120264	0.008727	
	CBJL	0.120264	0.008727	
	CBR	0.322025	0.044549	
	CBL	0.322025	0.044549	
	CPMBR	0.511457	0.462342	
	CPMBL	0.511457	0.462342	

*** TABULATED TOTAL COEFFICIENTS FOR COMPOSITE CASE 1 ***

ALPHA	CCL	CCL**2	CCM(MC)	CLL	CDITZ	CCT	CNI	CN	CCY
-10.000000	-0.3339382	0.1115147	-0.2692851	0.0	-0.0057847	1.0057840	0.0	0.0	0.0
-9.000000	-0.2375821	0.0564452	-0.2700175	0.0	-0.0052239	1.0052233	0.0	0.0	0.0
-8.000000	-0.1412260	0.0199448	-0.2707499	0.0	-0.0036565	1.0036554	0.0	0.0	0.0
-7.000000	-0.0448599	0.0020133	-0.2714822	0.0	-0.0010824	1.0010815	0.0	0.0	0.0
-6.000000	0.0514852	0.0026509	-0.2722145	0.0	0.0024982	0.9975014	0.0	0.0	0.0
-5.000000	0.1478423	0.0218573	-0.2729469	0.0	0.0070856	0.9929140	0.0	0.0	0.0
-4.000000	0.2441984	0.0596328	-0.2736793	0.0	0.0126796	0.9873200	0.0	0.0	0.0
-3.000000	0.3405545	0.1159773	-0.2744116	0.0	0.0192803	0.9807193	0.0	0.0	0.0
-2.000000	0.4369106	0.1908908	-0.2751440	0.0	0.0268876	0.9731120	0.0	0.0	0.0
-1.000000	0.5332667	0.2843733	-0.2758763	0.0	0.0355015	0.9644981	0.0	0.0	0.0
0.0	0.6296228	0.3964249	-0.2766087	0.0	0.0451221	0.9548775	0.0	0.0	0.0
1.000000	0.7259789	0.5270452	-0.2773410	0.0	0.0557494	0.9442502	0.0	0.0	0.0
2.000000	0.8223349	0.6762347	-0.2780734	0.0	0.0673832	0.9326164	0.0	0.0	0.0
3.000000	0.9186910	0.8439932	-0.2788057	0.0	0.0800238	0.9199758	0.0	0.0	0.0
4.000000	1.0150471	1.0303202	-0.2795381	0.0	0.0936710	0.9063286	0.0	0.0	0.0
5.000000	1.1114025	1.2352152	-0.2802705	0.0	0.1083248	0.8916748	0.0	0.0	0.0
6.000000	1.2077589	1.4566811	-0.2810028	0.0	0.1239854	0.8760143	0.0	0.0	0.0
7.000000	1.3041153	1.7007160	-0.2817352	0.0	0.1406525	0.8593472	0.0	0.0	0.0
8.000000	1.4004707	1.9613180	-0.2824675	0.0	0.1583263	0.8416733	0.0	0.0	0.0
9.000000	1.4968271	2.2404909	-0.2831998	0.0	0.1770068	0.8229929	0.0	0.0	0.0
10.000000	1.5931835	2.5382328	-0.2839322	0.0	0.1966938	0.8033058	0.0	0.0	0.0
11.000000	1.6895390	2.8545418	-0.2846646	0.0	0.2173876	0.7825120	0.0	0.0	0.0
12.000000	1.7858953	3.1894217	-0.2853969	0.0	0.2390880	0.7609116	0.0	0.0	0.0
13.000000	1.8822508	3.5428677	-0.2861293	0.0	0.2617950	0.7382046	0.0	0.0	0.0
14.000000	1.9786072	3.9148855	-0.2868617	0.0	0.2855088	0.7144909	0.0	0.0	0.0
15.000000	2.0749636	4.3054733	-0.2875940	0.0	0.3102291	0.6897705	0.0	0.0	0.0
16.000000	2.1713200	4.7146301	-0.2883264	0.0	0.3359562	0.6640435	0.0	0.0	0.0
17.000000	2.2676754	5.1423512	-0.2890587	0.0	0.3626897	0.6373099	0.0	0.0	0.0
18.000000	2.3640318	5.5885459	-0.2897910	0.0	0.3904300	0.6095696	0.0	0.0	0.0
19.000000	2.4603882	6.0533097	-0.2905234	0.0	0.4191770	0.5808226	0.0	0.0	0.0
20.000000	2.5567436	6.5369377	-0.2912558	0.0	0.4489306	0.5510691	0.0	0.0	0.0
21.000000	2.6531000	7.0389395	-0.2919881	0.0	0.4796908	0.5203089	0.0	0.0	0.0
22.000000	2.7494564	7.5595102	-0.2927205	0.0	0.5114577	0.4885419	0.0	0.0	0.0
23.000000	2.8458119	8.0986443	-0.2934529	0.0	0.5442312	0.4557684	0.0	0.0	0.0
24.000000	2.9421682	8.6563530	-0.2941852	0.0	0.5780115	0.4219882	0.0	0.0	0.0
25.000000	3.0385246	9.2326317	-0.2949175	0.0	0.6127983	0.3872014	0.0	0.0	0.0
26.000000	3.1348801	9.8274727	-0.2956499	0.0	0.6485918	0.3514079	0.0	0.0	0.0
27.000000	3.2312365	10.4408884	-0.2963822	0.0	0.6853919	0.3146077	0.0	0.0	0.0
28.000000	3.3275929	11.0724741	-0.2971146	0.0	0.7231987	0.2768009	0.0	0.0	0.0
29.000000	3.4239483	11.7234211	-0.2978470	0.0	0.7620121	0.2379875	0.0	0.0	0.0
30.000000	3.5203047	12.3925447	-0.2985793	0.0	0.8018323	0.1981574	0.0	0.0	0.0

```
*****************************************************
*  SECOND RUN FOR STABILITY DERIVATIVE CASE  *
*****************************************************

*********************************************************
*  STABILITY DERIVATIVE DATA FOR FUNDAMENTAL CASE  1  *
*********************************************************

ROLLING MOMENT COEFFICIENT DERIVATIVE DUE TO YAWING, CLLR =    0.0001168

YAWING MOMENT COEFFICIENT ABOUT XMC DUE TO YAWING ABOUT XCG, CN(R) MAY BE CALCULATED AS FOLLOWS

                    CN(R) = CNR*R + CNR2*R**2

                    WHERE   CNR =   0.000001049
                           CNR2 =   0.0

SIDE FORCE COEFFICIENT DUE TO YAWING, CY(R) MAY BE CALCULATED AS FOLLOWS

                    CY(R) = CYR*R + CYR2*R**2

                    WHERE   CYR =   0.0
                           CYR2 =   0.0

*********************************************************
*  STABILITY DERIVATIVE DATA FOR FUNDAMENTAL CASE  2  *
*********************************************************

ROLLING MOMENT COEFFICIENT DERIVATIVE DUE TO YAWING, CLLR =    0.0000436

YAWING MOMENT COEFFICIENT ABOUT XMC DUE TO YAWING ABOUT XCG, CN(R) MAY BE CALCULATED AS FOLLOWS

                    CN(R) = CNR*R + CNR2*R**2

                    WHERE   CNR =   0.000000034
                           CNR2 =   0.0

SIDE FORCE COEFFICIENT DUE TO YAWING, CY(R) MAY BE CALCULATED AS FOLLOWS

                    CY(R) = CYR*R + CYR2*R**2

                    WHERE   CYR =   0.0
                           CYR2 =   0.0

*********************************************************
*  STABILITY DERIVATIVE DATA FOR FUNDAMENTAL CASE  3  *
*********************************************************

ROLLING MOMENT COEFFICIENT DERIVATIVE DUE TO YAWING, CLLR =    0.0000344

YAWING MOMENT COEFFICIENT ABOUT XMC DUE TO YAWING ABOUT XCG, CN(R) MAY BE CALCULATED AS FOLLOWS

                    CN(R) = CNR*R + CNR2*R**2

                    WHERE   CNR =  -0.000000100
                           CNR2 =   0.0

SIDE FORCE COEFFICIENT DUE TO YAWING, CY(R) MAY BE CALCULATED AS FOLLOWS

                    CY(P) = CYR*R + CYR2*R**2

                    WHERE   CYR =   0.0
                           CYP2 =   0.0

*********************************************************
*  STABILITY DERIVATIVE DATA FOR FUNDAMENTAL CASE  4  *
*********************************************************

ROLLING MOMENT COEFF DERIVATIVE DUE TO ROLLING, CLLP =  -0.0066949

YAWING MOMENT COEFFICIENT ABOUT XMC DUE TO ROLLING, CN(P) MAY BE CALCULATED AS FOLLOWS

                    CN(P) = CNP2*P**2

                    WHERE  CNP2 =   0.0

SIDE FORCE COEFFICIENT DUE TO ROLLING, CY(P) MAY BE CALCULATED AS FOLLOWS

                    CY(P) = CYP2*P**2

                    WHERE  CYP2 =   0.0
```

STABILITY DERIVATIVE DATA FOR COMPOSITE CASE 1

FUNDAMENTAL CASE FACTORS

A(1)	A(2)	A(3)	A(4)	A(5)	A(6)	A(7)	A(8)	A(9)	A(10)
0.0	10.000000	10.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0

LIFT COEFFICIENT DERIVATIVE DUE TO PITCHING ABOUT XCG, CLQ = 0.089478
PITCHING MOMENT COEFFICIENT DERIVATIVE ABOUT ORIGIN DUE TO PITCHING ABOUT XCG, CMQ = -0.038299
PITCHING MOMENT COEFF DERIVATIVE ABOUT XMC DUE TO PITCHING ABOUT XCG, CMQMC = -0.015929

ROLLING MOMENT COEFF DERIVATIVE DUE TO ROLLING, CLLP = -0.0066949

YAWING MOMENT COEFFICIENT ABOUT XMC DUE TO ROLLING, CN(P) MAY BE CALCULATED AS FOLLOWS
CN(P) = CNP*P + CNP2*P**2
WHERE CNP = CNPO + CNPA*ALPHA
CNPO = -0.0003531
CNPA = -0.0000601
CNP2 = 0.0

SIDE FORCE COEFFICIENT DUE TO ROLLING, CY(P) MAY BE CALCULATED AS FOLLOWS
CY(P) = CYP*P + CYP2*P**2
WHERE CYP = CYPO + CYPA*ALPHA
CYPO = 0.0
CYPA = 0.0
CYP2 = 0.0

ROLLING MOMENT COEFF DERIVATIVE DUE TO YAWING ABOUT XCG, CLLR MAY BE CALCULATED AS FOLLOWS
CLLR = CLLRO + CLLRA*ALPHA
WHERE CLLRO = 0.0007805
CLLRA = 0.0001168

YAWING MOMENT COEFFICIENT ABOUT XMC DUE TO YAWING ABOUT XCG, CN(R) MAY BE CALCULATED AS FOLLOWS
CN(R) = CNR*R + CNR2*R**2
WHERE CNR = CNRO + CNRA*ALPHA + CNRA2*ALPHA**2
CNRO = -0.0000130
CNRA = -0.0000035
CNRA2 = 0.0000010
AND CNR2 = CNR2O + CNR2A*ALPHA + CNR2A2*ALPHA**2
CNR2O = 0.0
CNR2A = 0.0
CNR2A2 = 0.0

SIDE FORCE COEFFICIENT ABOUT XMC DUE TO YAWING ABOUT XCG, CY(R) MAY BE CALCULATED AS FOLLOWS
CY(R) = CYR*R + CYR2*R**2
WHERE CYR = CYRO + CYRA*ALPHA + CYRA2*ALPHA**2
CYRO = 0.0
CYRA = 0.0
CYRA2 = 0.0
AND CYR2 = CYR2O + CYR2A*ALPHA + CYR2A2*ALPHA**2
CYR2O = 0.0
CYR2A = 0.0
CYR2A2 = 0.0

VARIATION OF STABILITY TERMS WITH ANGLE OF ATTACK

ALPHA	CNP	CNP2	CYP	CYP2	CLLR	CNR	CNR2
-10.000000	0.0002479	0.0	0.0	0.0	-0.0003880	0.0001273	0.0
-9.000000	0.0001878	0.0	0.0	0.0	-0.0002712	0.0001038	0.0
-8.000000	0.0001277	0.0	0.0	0.0	-0.0001543	0.0000824	0.0
-7.000000	0.0000676	0.0	0.0	0.0	-0.0000375	0.0000632	0.0
-6.000000	0.0000075	0.0	0.0	0.0	0.0000794	0.0000460	0.0
-5.000000	-0.0000526	0.0	0.0	0.0	0.0001962	0.0000309	0.0
-4.000000	-0.0001127	0.0	0.0	0.0	0.0003131	0.0000179	0.0
-3.000000	-0.0001728	0.0	0.0	0.0	0.0004299	0.0000070	0.0
-2.000000	-0.0002329	0.0	0.0	0.0	0.0005468	-0.0000017	0.0
-1.000000	-0.0002930	0.0	0.0	0.0	0.0006636	-0.0000084	0.0
0.0	-0.0003531	0.0	0.0	0.0	0.0007805	-0.0000130	0.0
1.000000	-0.0004132	0.0	0.0	0.0	0.0008973	-0.0000155	0.0
2.000000	-0.0004733	0.0	0.0	0.0	0.0010142	-0.0000159	0.0
3.000000	-0.0005334	0.0	0.0	0.0	0.0011310	-0.0000142	0.0
4.000000	-0.0005935	0.0	0.0	0.0	0.0012479	-0.0000104	0.0
5.000000	-0.0006536	0.0	0.0	0.0	0.0013647	-0.0000045	0.0
6.000000	-0.0007137	0.0	0.0	0.0	0.0014815	0.0000035	0.0
7.000000	-0.0007738	0.0	0.0	0.0	0.0015984	0.0000136	0.0
8.000000	-0.0008339	0.0	0.0	0.0	0.0017152	0.0000258	0.0
9.000000	-0.0008940	0.0	0.0	0.0	0.0018321	0.0000401	0.0
10.000000	-0.0009541	0.0	0.0	0.0	0.0019489	0.0000565	0.0
11.000000	-0.0010142	0.0	0.0	0.0	0.0020658	0.0000750	0.0
12.000000	-0.0010743	0.0	0.0	0.0	0.0021826	0.0000956	0.0
13.000000	-0.0011344	0.0	0.0	0.0	0.0022995	0.0001182	0.0
14.000000	-0.0011945	0.0	0.0	0.0	0.0024163	0.0001430	0.0
15.000000	-0.0012546	0.0	0.0	0.0	0.0025332	0.0001699	0.0
16.000000	-0.0013147	0.0	0.0	0.0	0.0026500	0.0001989	0.0
17.000000	-0.0013748	0.0	0.0	0.0	0.0027669	0.0002299	0.0
18.000000	-0.0014349	0.0	0.0	0.0	0.0028837	0.0002631	0.0
19.000000	-0.0014950	0.0	0.0	0.0	0.0030006	0.0002984	0.0
20.000000	-0.0015551	0.0	0.0	0.0	0.0031174	0.0003358	0.0
21.000000	-0.0016152	0.0	0.0	0.0	0.0032343	0.0003752	0.0
22.000000	-0.0016753	0.0	0.0	0.0	0.0033511	0.0004168	0.0
23.000000	-0.0017354	0.0	0.0	0.0	0.0034680	0.0004605	0.0
24.000000	-0.0017955	0.0	0.0	0.0	0.0035848	0.0005062	0.0
25.000000	-0.0018556	0.0	0.0	0.0	0.0037017	0.0005541	0.0
26.000000	-0.0019157	0.0	0.0	0.0	0.0038185	0.0006040	0.0
27.000000	-0.0019758	0.0	0.0	0.0	0.0039354	0.0006561	0.0
28.000000	-0.0020359	0.0	0.0	0.0	0.0040522	0.0007102	0.0
29.000000	-0.0020960	0.0	0.0	0.0	0.0041690	0.0007665	0.0
30.000000	-0.0021561	0.0	0.0	0.0	0.0042859	0.0008248	0.0

THE PROGRAM HAS REACHED NORMAL TERMINATION

APPENDIX B - ERROR MESSAGES

In order to avoid wasting computing time, some types of errors or inconsistencies in input data can be checked by the program before beginning computation. Sometimes such errors make the program unable to continue execution, and sometimes the program can recover by ignoring the data or substituting correct data. In any case, an error message will be printed, enabling the user to identify the type of error or warning him that the input data may have been altered. Below is a list of the messages which may be printed and the subroutines in which they occur. At the end of each run a status message is always printed indicating whether the program has reached a normal or abnormal conclusion.

Message	Subroutine
** * THE PROGRAM HAS REACHED NORMAL TERMINATION * **	MAIN
******** ** * THE PROGRAM HAS REACHED ABNORMAL TERMINATION * ** ********	MAIN
THE ISYMM FLAG INDICATED AN ANTI-SYMETRIC CASE. HOWEVER, IT WILL BE TREATED AS SYMETRIC.	APPLY2
THE IGTYPE FLAG HAS THE VALUE OF** ONLY THE VALUES 1 OR 2 ARE ACCEPTABLE THIS CASE HAS BEEN TERMINATED	STAGE1
NROWS =***	STAGE1
NUMBER OF WING ROW TYPES =***	INPTS
*** WING ELEMENTS PRESCRIBED FOR ROW TYPE***	INPTS
NUMBER OF JET ROW TYPES =***	INPUTJ
*** JET ELEMENTS PRESCRIBED FOR ROW TYPE***	INPUTJ
3 ROW CONTINUITY RULE FAILURE	INPUTJ
AN INCONSISTENCY HAS BEEN FOUND IN THE SECTIONAL LEADING AND TRAILING EDGE INPUT	XLETR1
PLEASE CHECK YOUR SECTION LOCATION (Y) INPUT	BOXS
**** IS TOO MANY ELEMENTS	BOXS
A ZERO VALUE OF CMU HAS BEEN INPUT. THIS CMU CASE HAS BEEN IGNORED.	BOXJ
FUNDAMENTAL GEOMETRIC CASE*** AN INCONSISTENCY HAS BEEN FOUND IN THE HINGE INPUT DATA FOR WING ROW***, ROW TYPE***	BEECEE
AN INCORRECT COMPOSITE CASE INPUT VALUE HAS BEEN FOUND. IT WILL BE IGNORED.	INCOMP
MORE THAN [illegible] COMPOSITE CASES HAVE BEEN INPUT. SUBSEQUENT INPUTS WILL BE IGNORED.	INCOMP
AN END OF FILE HAS BEEN READ DURING COMPOSITE CASE INPUT	INCOMP
MATRIX DOES NOT HAVE ENOUGH CORE TO WORK THIS CASE HAS BEEN TERMINATED	STG2S

```
*DECK MAIN
      OVERLAY(OVLY,0,0)
      PROGRAM EVD(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     1            TAPE1,TAPE2,TAPE3,TAPE4)
C
C
C
C*************************************************************************
C*******************   EVD JET-WING COMPUTER PROGRAM   *******************
C*************************************************************************
C
C
C  THE DEVELOPMENT OF THE ELEMENTARY VORTEX DISTRIBUTION (EVD) JET-WING
C  COMPUTER PROGRAM FOR DETERMINING THE AERODYNAMIC CHARACTERISTICS OF
C  ARBITRARY JET FLAPPED WINGS WAS PERFORMED BY THE DOUGLAS AIRCRAFT
C  COMPANY, V/STOL TECHNOLOGY GROUP - AERODYNAMICS, OF THE MCDONNELL
C  DOUGLAS CORPORATION.   ORIGINALLY DEVELOPED UNDER THE SPONSORSHIP OF
C  THE INDEPENDENT RESEARCH AND DEVELOPMENT PROGRAM OF MCDONNELL
C  DOUGLAS, THE DOUGLAS EVD JET WING LIFTING SURFACE THEORY HAS BEEN
C  THE SUBJECT OF EXTENSIONS AND IMPROVEMENTS WHICH HAVE BEEN
C  ACCOMPLISHED UNDER OFFICE OF NAVAL RESEARCH CONTRACT N00014-71-C-0250
C  (T. I. WILSON - PROJECT ENGINEER).   WORK LEADING TO THE PRESENT
C  COMPUTER PROGRAM, WHICH INCORPORATES SEVERAL FEATURES ORIGINALLY
C  DEVELOPED BY THE DOUGLAS AIRCRAFT COMPANY, WAS ALSO CONDUCTED
C  UNDER THE SPONSORSHIP OF THIS CONTRACT.
C
C  IN SUMMARY, THE EVD JET WING COMPUTER PROGRAM WILL PROVIDE,
C  FOR ARBITRARY PLANFORMS, THE FOLLOWING -
C          1.  SPANWISE AND CHORDWISE LOADING
C          2.  SPANWISE VARIATION OF INDUCED DRAG
C          3.  A CAPABILITY TO INVESTIGATE THE EFFECTS OF -
C              A.  PART SPAN FLAPS
C              B.  PART SPAN BLOWING
C              C.  ROLLING, YAWING, PITCHING AND SIDESLIP
C          4.  TOTAL LIFT AND INDUCED DRAG (TREFFTZ PLANE METHOD),
C              PITCHING, YAWING AND ROLLING MOMENTS, ETC.
C
C  COMPLETE DOCUMENTATION OF THE EVD JET WING LIFTING SURFACE THEORY
C  AND ASSOCIATED COMPUTER PROGRAM ARE CONTAINED IN DOUGLAS REPORT
C
C            J5519 -- A THEORETICAL METHOD FOR CALCULATING THE
C        AERODYNAMIC CHARACTERISTICS OF ARBITRARY JET FLAP WINGS
C
C                              VOLUME I
C           THE ELEMENTARY VORTEX DISTRIBUTION JET-WING
C                       LIFTING SURFACE THEORY
C
C                              VOLUME II
C          EVD JET-WING COMPUTER PROGRAM USERS MANUAL
C
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON /LUKE/ TITLE(8)
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR
      COMMON/INDATA/ARE,SPA,CRE,XM,CMA,XC,NRO,NC,ISY,IPR,JET,IGT,IHI
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DFL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)
      COMMON/FCASE2/TST(40,10),HL(40,10),DJ(40),ACTE(40),AC(20,40),
     1             XHB(4,40),PET(4,40),IFS(4,40),ICT(40),IHT(40),NCT,NHT
      COMMON/FCASE3/EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10)
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)
      COMMON/SOLV1/B(600,10)
      COMMON/COMPOS/FACTOR(10,24),NCC
      COMMON/DERIV/UC(40),CLO,CMO,CMOMC
      COMMON/FINDIT/INDEX(1001)
      DATA CHECK/10H9        /
C
      CALL OPENMS(1,INDEX,1001,0)
C
C  READ THE TITLE FOR THIS CASE
   10 READ(5, 20 ) TITLE
   20 FORMAT(8A10)
      IF(EOF(5) .NE. 0) GO TO 110
C
C  READ GENERAL GEOMETRY CONTROL DATA
   30 READ(5, 40 ) AREA,SPAN,CREF,XMC,XCG
   40 FORMAT(5E10.6)
      READ(5, 50 ) NROWS,NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE,IDERIV
   50 FORMAT(10I2)
      ARE = AREA
      SPA = SPAN
      CRE = CREF
      XM  = XMC
      XC  = XCG
      NRO = NROWS
      NC  = NCASES
      ISY = ISYMM
      IPR = IPRINT
      JET = JETFLG
      IGT = IGTYPE
      IHI = IHINGE
C
C  FIND OUT WHICH TYPE OF RUN IS REQUIRED
      IF(IDERIV .NE. 0) GO TO 70
C
C  A REGULAR RUN WILL BE EXECUTED
   60 CALL APPLY1
      GO TO ( 70 , 80 , 110 , 130 ), IR
C
C  A STABILITY DERIVATIVE RUN WILL BE EXECUTED
   70 CALL APPLY2
      IF(IR .EQ. 2) GO TO 130
C
C*************************************************************************
C  THIS RUN HAS BEEN COMPLETED.  THANK GOD FOR SMALL BLESSINGS.
C*************************************************************************
C
C  PRINT COMPLETION MESSAGE FOR THIS RUN AND GO BACK TO BEGIN A NEW RUN
   80 WRITE(6, 90 )
   90 FORMAT(1H0/// 32X,10(5H*****), 3H***/ 32X,
     1        53H*   THE   PROGRAM   HAS   REACHED   NORMAL   TERMINATION   */
     2        32X,10(5H*****),3H***)
C  READ TO SEE IF THE NEXT CARD IS A TITLE OR AN OLD END OF CMU CARD
      READ(5, 20 ) TITLE
      IF(EOF(5) .NE. 0) GO TO 110
  100 IF(TITLE(1) .EQ. CHECK) GO TO 10
      GO TO 30
C
C  PRINT COMPLETION MESSAGE AND STOP EXECUTION
  110 WRITE(6, 90 )
  120 STOP
C
C*************************************************************************
C  A FATAL ERROR HAS OCCURED.  PRINT FINAL MESSAGE AND STOP EXECUTION.
C*************************************************************************
  130 WRITE(6, 140 )
  140 FORMAT(1H0///62X,2(4H****)/31X,11(5H*****)/
     1   31X,55H*   THE   PROGRAM   HAS   REACHED   ABNORMAL   TERMINATION   */
     2   31X,11(5H*****)/62X,2(4H****))
  150 STOP
      END
```

```
*DECK APPLY1
      SUBROUTINE APPLY1
C
C  THIS SUBROUTINE CONTROLS ALL ASPECTS OF CALCULATION OF REGULAR CASES
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR
C  DECIDE WHETHER OR NOT THERE IS AN ALPHA CASE
      NOALFA = 1
      IP = 1
      LOGIC = 1
      IF(ISYMM .LT. 0) NOALFA = 0
C
C  INITIALIZE AND INCREMENT THE CMU CASE CONTROL COUNTER
   10 NEWCMU = 0
   20 NEWCMU = NEWCMU + 1
C
C  EXECUTE THE PROBLEM FORMATION STAGE
   30 CALL OVERLAY(4HOVLY,1,0)
      GO TO ( 40 , 60 , 70 , 80 ), IR
C
C  EXECUTE THE PROBLEM SOLUTION STAGE
   40 CALL OVERLAY(4HOVLY,2,0)
      IF(IR .EQ. 2) GO TO 80
C
C  EXECUTE THE AERODYNAMIC PARAMETER STAGE
   50 CALL OVERLAY(4HOVLY,3,0)
C
C  THE PROGRAM HAS BEEN EXECUTED SUCESSFULLY
C  GO BACK AND DO A NEW CMU CASE
      IF(JETFLG .NE. 0) GO TO 60
      GO TO 20
C
C*************************************************************************
C  THIS RUN HAS BEEN COMPLETED.  RETURN TO START A NEW RUN.
   60 IR = 2
      RETURN
C  THIS RUN HAS BEEN COMPLETED.  NO FURTHER RUNS FOLLOW.
   70 IR = 3
      RETURN
C  A FATAL ERROR HAS OCCURED.  RETURN AND QUIT.
   80 IR = 4
      RETURN
C*************************************************************************
      END
```

```
*DECK APPLY2
      SUBROUTINE APPLY2
C
C   THIS SUBROUTINE CONTROLS ALL ASPECTS OF CALCULATION OF
C   STABILITY DERIVATIVES
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR
C
C   CHECK ON STATUS OF CONTROL FLAGS
   10 IHINGE = 0
      NOALFA =1
      NEWCMU = 1
      IF(ISYMM .GE. 0) GO TO 30
      ISYMM = 0
      WRITE(6, 20 )
   20 FORMAT(1H0///16X,41HTHE ISYMM FLAG INDICATED AN ANTI-SYMETRIC,
     1      48H CASE.  HOWEVER, IT WILL BE TREATED AS SYMETRIC.)
C
C   EXECUTE THE FIRST RUN
C
C   FORMULATE THE PROBLEM AS USUAL
   30 CALL OVERLAY(4HOVLY,1,0)
      GO TO ( 40 , 110 , 100 , 110 ), IR
C
C   ADD THE EXTRA FUNDAMENTAL CASE FOR DERIVATIVES DUE TO PITCHING
   40 LOGIC = 1
      CALL OVERLAY(4HOVLY,4,0)
C
C   EXECUTE THE PROBLEM SOLUTION STAGE AS USUAL FOR THE FIRST RUN
      LOGIC = 1
   50 CALL OVERLAY(4HOVLY,2,0)
      IF(IR .EQ. 2) GO TO 110
C
C   EXECUTE THE AERODYNAMIC STAGE FOR THE FIRST RUN
C   FUNDAMENTAL CASES
   60 LOGIC = 2
      CALL OVERLAY(4HOVLY,3,0)
C
C   THE FIRST RUN HAS BEEN COMPLETED
C   EXECUTE THE SECOND RUN
C
      WRITE(6, 70 )
   70 FORMAT(1H1////////////// 37X,11(4H****),2H** /
     1         37X,46H*   SECOND RUN FOR STABILITY DERIVATIVE CASE  * /
     2         37X,11(4H****),2H**)
C   IF THIS IS A SYMETRIC WING, SWITCH IT TO ANTI-SYMETRIC FOR RUN 2
   80 IF(ISYMM .EQ. 0) ISYMM = -1
C
C   STORE THE FIRST RUN SOLUTION ON UNIT 1, DEFINE THE FUNDAMENTAL CASES
C   FOR YAWING AND ROLLING RATES, AND PRINT THE NEW FUND CASE GEOMETRY.
   90 LOGIC = 2
      CALL OVERLAY(4HOVLY,4,0)
C
C   SET UP AND SOLVE THE MATRIX SYSTEM FOR THE SECOND RUN
      LOGIC = 2
      CALL OVERLAY(4HOVLY,2,0)
      IF(IR .EQ. 2) GO TO 110
C
C   CALCULATE AND PRINT THE DERIVATIVES FOR ALL FUNDAMENTAL
C   AND COMPOSITE CASES
      LOGIC = 3
      IF(IPRINT .GE. 0) IPRINT = 2
      CALL OVERLAY(4HOVLY,3,0)
C
C************************************************************************
C   THIS IS THE END OF THE LINE
  100 IR = 1
      RETURN
C
C   A FATAL ERROR HAS OCCURED.  RETURN ABNORMALLY TO MAIN.
  110 IR = 2
      RETURN
C************************************************************************
      END


*DECK STAGE1
      OVERLAY(OVLY,1,0)
      PROGRAM STAGE1
C
C   THIS SUBROUTINE READS THE GENERAL GEOMETRY PARAMETERS AND FLAGS, AND
C   CONTROLS THE CALLING OF THE SPECIALIZED GEOMETRY SUBROUTINES
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NEDWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR
C
C   CHECK WHETHER THIS IS THE FIRST CMU CASE
      IF(NEWCMU .GT. 1) GO TO 50
      IF((NROWS .GT. 40) .OR. (NROWS .LT. 3)) GO TO 80
C
C   SECTIONAL INPUT
   10 IF((IGTYPE .EQ. 1).OR.(IGTYPE .EQ. 2))CALL SGMAIN(NOALFA,IR)
      GO TO ( 20 , 40 , 100 ), IR
C
C   USER INPUT
C
C   PRINT ERROR MESSAGE BECAUSE IGTYPE HAS THE WRONG VALUE
   20 WRITE(6, 30 ) IGTYPE
   30 FORMAT(1H1///44X,22HTHE IGTYPE FLAG HAS THE VALUE OF,I2/
     1         44X,37HONLY THE VALUES 1 OR 2 ARE ACCEPTABLE//
     2         44X,29HTHIS CASE HAS BEEN TERMINATED)
      GO TO 100
C
C   READ THE COMPOSITE CASE REQUIREMENTS
   40 CALL INCOMP(NCASES,IR)
      IF(IR .EQ. 2) GO TO 100
C
C   READ THE CMU DATA
   50 CALL BLOWIN(JETFLG,IR)
      GO TO ( 60 , 110 , 120 ), IR
   60 CALL BOXJ(NEWMAX,IR)
      IF(IR .EQ. 2) GO TO 50
C
C   RETURN NORMALLY TO THE CONTROL PROGRAM
   70 IR = 1
      GO TO 130
C
C   PRINT ERROR MESSAGE BECAUSE THE NROWS VALUE IS UNACCEPTABLE
   80 WRITE(6, 90 ) NROWS
   90 FORMAT(1H1/55X,7HNROWS =,I3)
C
C************************************************************************
C   A FATAL ERROR HAS OCCURED.   RETURN ABNORMALLY TO MAIN.
  100 IR = 4
      GO TO 130
C
C   THIS RUN HAS BEEN COMPLETED.   THANK GOD FOR SMALL BLESSINGS.
C
C   RETURN TO MAIN AND BEGIN A COMPLETELY NEW RUN
  110 IR = 2
      GO TO 130
C
C   RETURN TO MAIN AND STOP THE EXECUTION
  120 IR = 3
  130 CONTINUE
C************************************************************************
      END


*DECK SGMAIN
      SUBROUTINE SGMAIN(NOALFA,IR)
C
C   THIS SUBROUTINE CONTROLS ALL GEOMETRY CALCULATIONS FOR THE
C   SECTIONAL GEOMETRY METHOD
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
C
C   READ THE WING PLANFORM GEOMETRY DATA
   10 CALL INPTS(IR)
      IF(IR .EQ. 2) GO TO 100
      IF(IGTYPE .EQ. 1) CALL XLETR1(IR)
      IF(IR .EQ. 2) GO TO 100
      IF(IGTYPE .EQ. 2) CALL XLETR2
C
C   NORMALIZE THE WING PLANFORM GEOMETRY DATA
   20 CALL NORM1
C
C   READ THE JET SHEET GEOMETRY DATA
   30 CALL INPUTJ(IR)
      IF(IR .EQ. 2) GO TO 100
C
C   CONSTRUCT THE EVD ELEMENTS
   40 CALL BOXS(IR)
      IF(IR .EQ. 2) GO TO 100
C
C   CONSTRUCT THE SET OF FUNDAMENTAL GEOMETRIC CASES
C
      DO 90 N = 1,NCASES
      LCASE = N
C
C   READ THE GEOMETRY FOR THIS CASE
   50 CALL INCASE(LCASE,NOALFA)
C
C   CONSTRUCT THE CASE DATA
   60 CALL BFECFF(LCASE,NOALFA,IR)
      IF(IR .EQ. 2) GO TO 100
C
C   PRINT THE GEOMETRY AND CONSTRUCTED CASE DATA IF REQUIRED
      IF(LCASE .EQ. 1) WRITE(6, 70 )
   70 FORMAT(1H1)
   80 CALL OUT1(LCASE)
   90 CONTINUE
      IR = 2
      RETURN
C
C   AN ERROR HAS OCCURED.   RETURN ABNORMALLY TO STAGE1.
C
  100 IR = 3
      RETURN
      END
```

C2

```
*DECK INPUTS
      SUBROUTINE INPTS(IR)
C
C  THIS SUBROUTINE READS THE WING GEOMETRY DATA
C  FOR THE SECTIONAL GEOMETRY METHOD
C
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)
      COMMON/SG1/XBW(20,10),XBJ(20,10),ICTYPE(40),IJTYPE(40),
     1           NWTYPE,NJTYPE
      DIMENSION NI(10)
C
C  READ THE SECTIONAL PLANFORM DATA
   10 NWTYPE = 0
      READ(5, 20 ) (Y(K),K=1,NROWS)
   20 FORMAT(8F10.6)
      READ(5, 30 ) (ICTYPE(K),K=1,NROWS)
   30 FORMAT(40I2)
      DO 40 K = 1,NROWS
      IF(ICTYPE(K) .GT. NWTYPE) NWTYPE = ICTYPE(K)
   40 CONTINUE
      IF(NWTYPE .GT. 10) GO TO 80
      READ(5, 30 )(NI(N),N=1,NWTYPE)
C
C  READ THE CHORDWISE DIVISION DATA FOR EACH ROW TYPE
      DO 50 N = 1,NWTYPE
      NIN = NI(N)
      IF((NIN .LT. 1) .OR. (NIN .GT. 20)) GO TO 100
      READ(5, 20 ) (XBW(L,N),L=1,NIN)
   50 CONTINUE
C  DEFINE THE NUMBER OF CHORDWISE DIVISIONS FOR EACH ROW
      DO 70  K = 1,NROWS
      ICK = ICTYPE(K)
   60 NW(K) = NI(ICK)
   70 CONTINUE
      IR = 1
      RETURN
C
C  AN ERROR HAS OCCURED.    PRINT A MESSAGE AND QUIT.
   80 WRITE(6, 90 ) NWTYPE
   90 FORMAT(1H1/45X,26HNUMBER OF WING ROW TYPES =,I3)
      IR = 2
      RETURN
  100 WRITE(6, 110 ) NIN,N
  110 FORMAT(1H1/38X,I3,38H WING ELEMENTS PRESCRIBED FOR ROW TYPE,I3)
      IR = 2
      RETURN
      END


*DECK INPUTJ
      SUBROUTINE INPUTJ(IR)
C
C  THIS SUBROUTINE READS THE JET GEOMETRY INPUT
C  FOR THE SECTIONAL GEOMETRY METHOD
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/SG1/XBW(20,10),XBJ(20,10),ICTYPE(40),IJTYPE(40),
     1           NWTYPE,NJTYPE
      DIMENSION NI(10)
C
C  READ THE TYPE OF DIVISION FOR EACH ROW
   10 NJTYPE = 0
      NROWSJ = 0
      IF(JETFLG .NE. 0) GO TO 110
      READ(5, 20 ) (IJTYPE(K),K=1,NROWS)
   20 FORMAT(40I2)
      DO 30 K = 1,NROWS
      IF(IJTYPE(K) .GT. NJTYPE) NJTYPE = IJTYPE(K)
      IF(IJTYPE(K) .NE. 0) NROWSJ = NROWSJ + 1
   30 CONTINUE
      IF(NJTYPE .GT. 10) GO TO 130
C  READ THE NUMBER OF CHORDWISE DIVISIONS IN EACH ROW TYPE
      READ(5, 40 ) (NI(N),N=1,NJTYPE)
   40 FORMAT(10I2)
C  READ THE CHORDWISE DIVISION DATA FOR EACH ROW TYPE
      DO 60 N = 1,NJTYPE
      NIN = NI(N)
      IF((NIN .LT. 1) .OR. (NIN .GT. 20)) GO TO 150
      READ(5, 50 ) (XBJ(L,N),L=1,NIN)
   50 FORMAT(8F10.6)
   60 CONTINUE
C
C  DEFINE THE NUMBER OF CHORDWISE DIVISIONS FOR EACH ROW
      DO 80  K = 1,NROWS
      NJ(K) = 0
      IF(IJTYPE(K) .EQ. 0) GO TO 80
      IJK = IJTYPE(K)
   70 NJ(K) = NI(IJK)
   80 CONTINUE
C  CHECK FOR ROW CONSISTENCY ON EITHER SIDE OF JET
      ICOUNT = 1
      IF(NJ(1) .EQ. 0)ITEST = 0
      IF(NJ(1) .GT. 0)ITEST = 1
      DO 100 K = 1,NROWS
      IF(NJ(K) .EQ. 0)ICOMP = 0
      IF(NJ(K) .GT. 0)ICOMP = 1
      IF(ICOMP .EQ. ITEST)GO TO 90
      IF(ICOUNT .LT. 3)GO TO 170
      ICOUNT = 1
      IF(NJ(K) .EQ. 0)ITEST = 0
      IF(NJ(K) .GT. 0)ITEST = 1
      GO TO 100
   90 ICOUNT = ICOUNT + 1
  100 CONTINUE
      IR = 1
      RETURN
C
C  THERE IS NO JET FOR THIS RUN
  110 DO 120 K = 1,NROWS
      IJTYPE(K) = 0
      NJ(K) = 0
  120 CONTINUE
      IR = 1
      RETURN
C
C  AN ERROR HAS OCCURED.    PRINT A MESSAGE AND QUIT.
  130 WRITE(6, 140 ) NJTYPE
  140 FORMAT(1H1/25HNUMBER OF JET ROW TYPES =,I3)
      IR = 2
      RETURN
  150 WRITE(6, 160 ) NIN,N
  160 FORMAT(1H1/39X,I3,37H JET ELEMENTS PRESCRIBED FOR ROW TYPE,I3)
      IR = 2
      RETURN
  170 WRITE(6, 180 )
  180 FORMAT(1H1,29H3 ROW CONTINUITY RULE FAILURE)
      IR = 2
      RETURN
      END


*DECK XLETR1
      SUBROUTINE XLETR1(IR)
C
C  THIS SUBROUTINE READS THE LEADING AND TRAILING EDGE COORDINATES
C  AT SPANWISE STATIONS CONNECTED BY STRAIGHT LEADING AND TRAILING EDGES
C  AND INTERPOLATES TO GET THE COORDINATES AT INTERMEDIATE SECTIONS
C
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)
      DIMENSION YP(40),XLE(40),XTR(40)
C
C  READ XLEAD AND XTRAIL
      NX = 0
      DO 30 N = 1,NROWS
      READ(5, 10 )YP(N),XLE(N),XTR(N)
   10 FORMAT(3F10.6)
   20 IF(YP(N) .GT. 1.1) GO TO 40
      NX = NX + 1
   30 CONTINUE
C  CHECK WHETHER THE Y VALUES ARE REALISTIC
   40 IF(ABS(YP(1)-Y(1)) .GT. 0.0001)GO TO 110
      IF(ABS(YP(NX)-Y(NROWS)) .GT. 0.0001)GO TO 110
C
C  READ THE EXTRA 9 CARD IF NROWS CARDS HAVE BEEN INPUT
      IF(NX .EQ. NROWS) READ(5, 10 ) EXTRA9
C
C  INTERPOLATE FOR XLEAD AND XTRAIL AT THE INTERMEDIATE SECTIONS
      K = 0
      DO 100 N = 1,NX
   50 K = K + 1
      IF(K .GT. NROWS) GO TO 110
      IF(ABS(YP(N)-Y(K)) .GT. 0.0001)GO TO 70
C  XLE AND XTR WERE INPUT FOR ROW K
   60 XLEAD(K) = XLE(N)
      XTRAIL(K) = XTR(N)
      GO TO 100
C  XLE AND XTR MUST BE INTERPOLATED FOR ROW K
   70 NM1 = N - 1
   80 YRATIO  = (Y(K)-YP(N)) / (YP(NM1)-YP(N))
      XLEAD(K) = XLE(N) + YRATIO * (XLE(NM1)-XLE(N))
   90 XTRAIL(K) = XTR(N) + YRATIO * (XTR(NM1)-XTR(N))
      GO TO 50
  100 CONTINUE
      IR = 1
      RETURN
C
C  AN ERROR HAS OCCURED.    PRINT A MESSAGE AND RETURN.
  110 WRITE(6, 120 )
  120 FORMAT(1H1////20X,38HAN INCONSISTENCY HAS BEEN FOUND IN THE,
     1       42H SECTIONAL LEADING AND TRAILING EDGE INPUT)
      IR = 2
      RETURN
      END
```

```
*DECK XLETR2
      SUBROUTINE XLETR2
C
C   THIS SUBROUTINE READS THE FUNDAMENTAL PLANFORM PARAMETERS FOR A
C   TRAPEZOIDAL WING, AND CALCULATES THE LEADING AND TRAILING EDGE
C   COORDINATES AT EACH Y STATION.  NOTE THAT THE PLANFORM OUTLINE
C   MUST BE SYMETRIC.
C
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)
C
C   READ THE FUNDAMENTAL PLANFORM PARAMETERS
      READ(5, 10 ) ARATIO,SWEEP,TR
   10 FORMAT(4F10.6)
C
C   COMPUTE THE GENERAL PLANFORM CHARACTERISTICS
      B2 = SPAN / 2.00
      SW = SWEEP / 57.295779
   20 CROOT = 2.0 * SPAN / ((1.0+TR)*ARATIO)
      AREA = (1.0+TR) * CROOT * B2
      XLB2 = 0.250 * (1.0-TR) * CROOT + B2 * TAN(SW)
   30 CMAC = 2.0 * CROOT * (1.0 + TR + TR*TR) / (3.0*(1.0+TR))
      IF(CREF .EQ. 0.0) CREF = CMAC
      CBAR=AREA/SPAN
C
C   COMPUTE THE LEADING AND TRAILING EDGE COORDINATES
   40 DO 60 K = 1,NROWS
      YBAR = Y(K)
      IF(YBAR .LT. 0.0) YBAR = -YBAR
      XLEAD(K) = XLB2 * YBAR
   50 C = CROOT * (1.0-(1.0-TR)*YBAR)
      XTRAIL(K) = XLEAD(K) + C
   60 CONTINUE
      RETURN
      END


*DECK NORM1
      SUBROUTINE NORM1
C
C   THIS SUBROUTINE NORMALIZES ALL WING PLANFORM GEOMETRY BY SPAN/2
C
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)
   10 B2 = SPAN / 2.00
      AREA = AREA / B2**2
      CREF = CREF / B2
   20 XMC = XMC / B2
      XCG = XCG / B2
      DO 40 K = 1,NROWS
   30 XLEAD(K) = XLEAD(K) / B2
      XTRAIL(K) = XTRAIL(K) / B2
   40 CONTINUE
      SPAN = 2.00
      ARATIO = SPAN * SPAN / AREA
      RETURN
      END


*DECK BOXS
      SUBROUTINE BOXS(IR)
C
C   THIS SUBROUTINE CONSTRUCTS THE GEOMETRIC PARAMETERS FOR ALL THE
C   EVD ELEMENTS ON THE WING AND JET
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)
      COMMON/SG1/XBW(20,10),XBJ(20,10),ICTYPE(40),IJTYPE(40),
     1             NWTYPE,NJTYPE
C
C   CONSTRUCT THE ELEMENTS ON THE WING
C
C   COMPUTE SECTIONAL DATA
   10 CHORD(1) = XTRAIL(1) - XLEAD(1)
      DELTA(1) = 1.00 - Y(1)
      CMAC = CHORD(1)**2 * DELTA(1)
      DO 30 K = 2,NROWS
   20 CHORD(K) = XTRAIL(K) - XLEAD(K)
      DELTA(K) = Y(K-1) - Y(K) - DELTA(K-1)
      IF(DELTA(K) .LT. 0.0) GO TO 190
      CMAC = CMAC + CHORD(K)**2 * DELTA(K)
   30 CONTINUE
C   CHECK THE VALIDITY OF THE SECTIONAL ALLIGNMENT
      YD = Y(NROWS) - DELTA(NROWS)
      IF((ISYMM .GE. 0) .AND. (ABS(YD) .GT. 0.0001)) GO TO 190
      IF((ISYMM .EQ. 1) .AND. (ABS(YD+1.0) .GT. 0.0001)) GO TO 190
      DSUM = DELTA(1)
      DO 35 K = 2,NROWS
      YL = Y(K) + DELTA(K)
      YR = Y(K-1) - DELTA(K-1)
      IF(ABS(YR-YL) .GT. 0.0001) GO TO 190
      DSUM = DSUM + DELTA(K)
   35 CONTINUE
      IF(ABS(DSUM-0.50) .GT. 0.0001) GO TO 190
      CMAC = 2.0 * CMAC / AREA
      IF(ISYMM .LT. 1) CMAC = 2.0 * CMAC
      IF(CREF .LT. 0.0001) CREF = CMAC
      CALL TANS(TANLE,XLEAD,Y,NROWS)
C
C   COMPUTE ALL CHORDWISE ELEMENT PARAMETERS FOR EACH SECTION
      I = 0
      DO 90  K = 1,NROWS
C   COMPUTE X-COORDINATES
      NWK = NW(K)
      DO 50 L = 1,NWK
      I = I + 1
      ICK = ICTYPE(K)
   40 XB(I) = XBW(L,ICK)
   50 CONTINUE
C   COMPUTE ALL OTHER PARAMETERS
      I = I - NWK
      IW(K) = I + 1
      DO 80 L = 1,NWK
      I = I + 1
   60 KK(I) = K
      DEL(I) = XB(I+1) - XB(I)
   70 XI(I) = XLEAD(K) + XB(I) * CHORD(K)
      ITYPE(I) = 10
   80 CONTINUE
C   REDEFINE THE LAST DEL IN THIS SECTION, AND DEFINE THE L.E. EVD TYPE
      DEL(I) = 1.00 - XB(I)
      IWK = IW(K)
      ITYPE(IWK) = 20
   90 CONTINUE
      NWT = I
C
C   CONSTRUCT THE ELEMENTS ON THE JET SHEET
C
C   COMPUTE ALL CHORDWISE ELEMENT PARAMETERS FOR EACH SECTION
      IF(JETFLG .NE. 0) GO TO 180
      DO 170 K = 1,NROWS
C   COMPUTE X-COORDINATES
      IJ(K) = 0
  100 NJK = NJ(K)
      IF(NJK .EQ. 0) GO TO 170
      DO 120 L = 1,NJK
      I = I + 1
      IJK = IJTYPE(K)
  110 XB(I) = XBJ(L,IJK)
  120 CONTINUE
C   COMPUTE ALL OTHER PARAMETERS
      I = I - NJK
  130 IJ(K) = I + 1
      DO 160 L = 1,NJK
      I = I + 1
  140 KK(I) = K
      DEL(I) = XB(I+1) - XB(I)
  150 XI(I) = XLEAD(K) + XB(I) * CHORD(K)
      ITYPE(I) = 10
  160 CONTINUE
C   REDEFINE THE LAST DEL AND EVD TYPE AND THE D VALUE FOR THIS SECTION
      DEL(I) = 1.0E10
      ITYPE(I) = 30
      D(K) = XI(I) - XTRAIL(K)
  170 CONTINUE
  180 NMAX = I
      IF(NMAX .GT. 600) GO TO 210
      NJT = NMAX - NWT
      IR = 1
      RETURN
C
C   AN ERROR HAS OCCURED.   PRINT A MESSAGE AND QUIT.
  190 WRITE(6, 200 )
  200 FORMAT(1H1/38X,44HPLEASE CHECK YOUR SECTION LOCATION (Y) INPUT)
      IR = 2
      RETURN
  210 WRITE(6, 220 ) NMAX
  220 FORMAT(1H1/48X,I4,21H IS TOO MANY ELEMENTS)
      IR = 2
      RETURN
      END


*DECK TANS
      SUBROUTINE TANS(TAN,X,Y,NROWS)
C
C   THIS SUBROUTINE COMPUTES THE TANGENT OF THE LEADING OR TRAILING EGDE
C   SWEEP ANGLE AT THE CENTERLINE OF EACH SECTION.  IT IS ACCURATE FOR
C   SECTIONS WITH STRAIGHT EDGES IN GROUPS OF THREE OR MORE.
C   IT IS ONLY APPROXIMATE FOR CURVED EDGES.
C   IT MAY RESULT IN ERRORS FOR SECTIONS ADJACENT TO WING BREAKS,
C   IF STRAIGHT EDGES ARE IN ADJACENT GROUPS OF ONLY ONE OR TWO.
C
      DIMENSION TAN(40),X(40),Y(40),S(40)
      SLOP(XR,XL,YR,YL) = (XR-XL) / (YR-YL)
C
      DO 20 K = 1,NROWS
      KR = K-1
      KL = K
      IF(K .GT. 1) GO TO 10
      KR = 1
      KL = 2
   10 S(K) = SLOP(X(KR),X(KL),Y(KR),Y(KL))
   20 CONTINUE
      DO 50  K = 1,NROWS
      IF(K .LT. 3) GO TO 30
      IF(K .EQ. NROWS) GO TO 30
      IF(K .EQ. (NROWS-1)) GO TO 40
C   CHECK WHETHER THE RIGHT OR LEFT SIDES ARE STRAIGHT
      IF(ABS(S(K) - S(K-1)) .LT. 0.001) GO TO 30
      IF(ABS(S(K+1) - S(K+2)) .LT. 0.001) GO TO 40
C   NEITHER SIDE IS CONCLUSIVELY STRAIGHT - CHECK FURTHER LEFT AND RIGHT
      IF(K .EQ. 3) GO TO 40
      IF(K .EQ. (NROWS-2)) GO TO 30
      IF(ABS(S(K-1) - S(K-2)) .LT. 0.001) GO TO 40
      IF(ABS(S(K+2) - S(K+3)) .LT. 0.001) GO TO 30
```

```
C  THE TRUE SHAPE CANNOT BE DETERMINED - GIVE UP AND TAKE THE AVERAGE
      TAN(K) = (S(K) + S(K+1)) / 2.00
      GO TO 50
C  THE RIGHT EDGE IS STRAIGHT
   30 TAN(K) = S(K)
      GO TO 50
C  THE LEFT EDGE IS STRAIGHT
   40 TAN(K) = S(K+1)
   50 CONTINUE
      RETURN
      END


*DECK INCASE
      SUBROUTINE INCASE(LCASE,NOALFA)
C
C  THIS SUBROUTINE READS THE FUNDAMENTAL GEOMETRIC CASE DATA
C
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/FCASE1/INPUTT,INPUTH,INPUTD,INPUTC,INPUTR
      COMMON/FCASE2/TST(40,10),HL(40,10),DJ(40),ACTE(40),AC(20,40),
     1              XHB(4,40),BET(4,40),IFS(4,40),ICT(40),IHT(40),NCT,NHT
      DIMENSION NI(10),DUMMY(40)
C
C  INITIALIZE SECTIONAL DATA
      DO 30 K = 1,NROWS
   10 TST(K,LCASE) = 0.00
      HL(K,LCASE) = 0.00
      DJ(K) = 0.00
      ACTE(K) = 0.00
      ICT(K) = 0
      IHT(K) = 0
C  INITIALIZE THE CAMBER ANGLES
      NWK = NW(K)
      DO 20 L = 1,NWK
      AC(L,K) = 0.00
   20 CONTINUE
   30 CONTINUE
C  INITIALIZE THE HINGE DATA
      DO 50 N = 1,NROWS
      DO 40 L = 1,4
      XHB(L,N) = 0.00
      BET(L,N) = 0.00
   40 CONTINUE
   50 CONTINUE
C
      IF((LCASE .EQ. 1) .AND. (NOALFA .GT. 0)) RETURN
C
C  READ FUNDAMENTAL CASE CONTROL FLAGS
      READ(5, 60 ) INPUTT,INPUTH,INPUTD,INPUTC,INPUTR
   60 FORMAT(5I2)
C
C  READ SECTIONAL TWIST, HEIGHT AND JET DEFLECTION DATA
      IF(INPUTT .NE. 0) READ(5, 70 ) (TST(K,LCASE),K=1,NROWS)
   70 FORMAT(8F10.6)
      IF(INPUTH .NE. 0) READ(5, 70 ) (HL(K,LCASE),K=1,NROWS)
      IF(INPUTD .EQ. 0) GO TO 90
      READ(5, 70 ) (DUMMY(K),K=1,NROWSJ)
C  DISTRIBUTE THE DUMMY VALUES PROPERLY IN THE DJ ARRAY
      KP = 0
      DO 80 K = 1,NROWS
      IF(NJ(K) .EQ. 0) GO TO 80
      KP = KP + 1
      DJ(K) = DUMMY(KP)
   80 CONTINUE
C
C  READ THE CAMBER ANGLES, IN DEGREES
   90 IF(INPUTC .EQ. 0) GO TO 160
      READ(5, 100 ) (ICT(K),K=1,NROWS)
  100 FORMAT(40I2)
      NCT = 0
      DO 110 K = 1,NROWS
      IF(ICT(K) .EQ. 0) GO TO 110
      IF(ICT(K) .GT. NCT) NCT = ICT(K)
      ICK = ICT(K)
      NI(ICK) = NW(K)
  110 CONTINUE
      DO 130 N = 1,NCT
      NIN = NI(N)
  120 READ(5, 70 ) (AC(L,N),L=1,NIN)
  130 CONTINUE
  140 IF(NROWSJ .GT. 0) READ(5, 70 ) (DUMMY(K),K=1,NROWSJ)
C  DISTRIBUTE THE DUMMY VALUES PROPERLY IN THE ACTE ARRAY
      KP = 0
      DO 150 K = 1,NROWS
      IF(NJ(K) .EQ. 0) GO TO 150
      KP = KP + 1
      ACTE(K) = DUMMY(KP)
  150 CONTINUE
C
C  READ THE HINGE LOCATION, TYPE AND TURNING ANGLE DATA
  160 IF(INPUTR .EQ. 0) GO TO 210
  170 READ(5, 100 ) (IHT(K),K=1,NROWS)
      NHT = 0
      DO 180 K = 1,NROWS
      IF(IHT(K) .GT. NHT) NHT = IHT(K)
  180 CONTINUE
      DO 200 N = 1,NHT
      READ(5, 190 ) (XHB(L,N),IFS(L,N),BET(L,N),L=1,4)
  190 FORMAT(4(F10.6,I1,F9.6))
  200 CONTINUE
  210 RETURN
      END
```

```
*DECK BECEFF
      SUBROUTINE BECEFF(LCASE,NOALFA,IR)
C
C  THIS SUBROUTINE CONSTRUCTS THE BOUNDARY CONDITION ARRAYS FOR THE
C  FUNDAMENTAL GEOMETRIC CASES
C
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1              D(40),KK(600),ITYPE(600)
      COMMON/FCASE1/INPUTT,INPUTH,INPUTD,INPUTC,INPUTR
      COMMON/FCASE2/TST(40,10),HL(40,10),DJ(40),ACTE(40),AC(20,40),
     1              XHB(4,40),BET(4,40),IFS(4,40),ICT(40),IHT(40),NCT,NHT
      COMMON/FCASE3/EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10)
      DIMENSION NH(40)
C
C  INITIALIZE THE BOUNDARY CONDITION ANGLES
      DO 20 I = 1,NMAX
   10 EPS(I,LCASE) = 0.00
      BETA(I,LCASE) = 0.00
   20 CONTINUE
      DO 30 K = 1,NROWS
      THETA(K,LCASE) = 0.00
      THS(K,LCASE) = 0.00
   30 CONTINUE
C
C  DEFINE THE ANGLES FOR THE ANGLE-OF-ATTACK FUNDAMENTAL CASE
      IF((LCASE .GT. 1) .OR. (NOALFA .EQ. 0)) GO TO 60
      DO 40 I = 1,NWT
      EPS(I,1) = 1.00
   40 CONTINUE
      DO 50 K = 1,NROWS
      IF(NJ(K) .GT. 0) THETA(K,1) = 1.000
   50 CONTINUE
      IR = 1
      RETURN
C
C  DEFINE THE ANGLES FOR ALL REMAINING FUNDAMENTAL CASES
C
C  CAMBER CONTRIBUTION
   60 IF(INPUTC .EQ. 0) GO TO 110
      I = 0
      DO 100 K = 1,NROWS
      IF(NJ(K) .EQ. 0) GO TO 70
      THETA(K,LCASE) = ACTE(K)
   70 NWK = NW(K)
      IF(ICT(K) .EQ. 0) GO TO 90
      DO 80 L = 1,NWK
      I = I + 1
      ICK = ICT(K)
      EPS(I,LCASE) = AC(L,ICK)
   80 CONTINUE
      GO TO 100
   90 I = I + NWK
  100 CONTINUE
C
C  TWIST CONTRIBUTION
  110 IF(INPUTT .EQ. 0) GO TO 160
      I = 0
      DO 150 K = 1,NROWS
      IF(NJ(K) .EQ. 0) GO TO 120
      THETA(K,LCASE) = THETA(K,LCASE) + TST(K,LCASE)
  120 NWK = NW(K)
      DO 140 L = 1,NWK
      I = I + 1
  130 EPS(I,LCASE) = EPS(I,LCASE) + TST(K,LCASE)
  140 CONTINUE
  150 CONTINUE
C
C  FLAP AND SLAT DEFLECTION CONTRIBUTION
C
  160 IF(INPUTR .EQ. 0) GO TO 320
C
C  SUM UP THE TOTAL SLAT ANGLE AND FIND THE NUMBER OF HINGES ON EACH ROW
      DO 190 K = 1,NROWS
      NH(K) = 0
      IF(IHT(K) .EQ. 0) GO TO 190
      DO 180 L = 1,4
  170 N = IHT(K)
      IF(XHB(L,N) .LT. 0.001) GO TO 180
      NH(K) = NH(K) + 1
      IF(IFS(L,N) .GT. 0) THS(K,LCASE) = THS(K,LCASE) + BET(L,N)
  180 CONTINUE
  190 CONTINUE
C
C  COMPUTE INCIDENCE OF EACH ELEMENT AND FIND TURNING ANGLE AND EVD TYPE
C  FOR EACH HINGE ELEMENT
      I = 0
      DO 310 K = 1,NROWS
      NWK = NW(K)
      N = IHT(K)
  200 IF(N .EQ. 0) GO TO 300
      LSTART = 1
      B = 0.0
      NHK = NH(K)
  210 IF(NHK .EQ. 0) GO TO 360
C  CYCLE THE HINGE POINTS IN CHORDWISE ORDER
      DO 270 LH = 1,NHK
C  CYCLE THE VORTEX POINTS IN CHORDWISE ORDER, LOOKING FOR NEXT HINGE
      DO 250 L = LSTART,NWK
      I = I + 1
C  CHECK ON RELATIVE LOCATION OF VORTEX POINT AND HINGE POINT
  220 XDIFF = XHB(LH,N) - XB(I)
      IF(ABS(XDIFF) .LT. 0.001) GO TO 230
      IF(XDIFF .GT. 0.001) GO TO 240
      IF(XDIFF .LT. -0.001) GO TO 360
C  THE ITH VORTEX POINT IS A HINGE POINT
  230 B = B + BET(LH,N)
      BETA(I,LCASE) = BET(LH,N)
```

C5

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

AD-A058 821 DOUGLAS AIRCRAFT CO LONG BEACH CALIF F/G 20/4
A THEORETICAL METHOD FOR CALCULATING THE AERODYNAMIC CHARACTERI--ETC(U)
1977 M L LOPEZ, C SHEN, N F WASSON N00014-71-C-0250

UNCLASSIFIED MDC-J5519-02-VOL-2 NL

2 OF 2

AD
A058 821

END
DATE
FILMED
11-78
DDC

```
      EPS(I,LCASE) = EPS(I,LCASE) - THS(K,LCASE) + B
      ITYPE(I) = 42
      IF(IES(LH,N) .GT. 0) ITYPE(I) = 41
      GO TO 260
C   THE ITH VORTEX POINT IS NOT A HINGE POINT
  240 EPS(I,LCASE) = EPS(I,LCASE) - THS(K,LCASE) + B
  250 CONTINUE
  260 LSTART = I - IW(K) + 1
  270 CONTINUE
C   DEFINE THE INCIDENCE ANGLE FOR REMAINING POINTS BEHIND THE LAST HINGE
      IF(LSTART .EQ. NWK) GO TO 290
      LSTART = LSTART + 1
      DO 280 L = LSTART,NWK
      I = I + 1
      EPS(I,LCASE) = EPS(I,LCASE) - THS(K,LCASE) + B
  280 CONTINUE
C   COMPUTE THE EFFECT OF THE HINGES ON THE JET ANGLE
  290 IF(NJ(K) .GT. 0) THETA(K,LCASE) = THETA(K,LCASE)-THS(K,LCASE)+B
      GO TO 310
  300 I = I + NWK
  310 CONTINUE
C
C   JET DEFLECTION CONTRIBUTION
C
  320 IF(INPUTD .EQ. 0) GO TO 350
      DO 340 K = 1,NROWS
      IF(NJ(K) .EQ. 0) GO TO 340
      I = IJ(K)
  330 BETA(I,LCASE) = DJ(K)
      IF(ABS(DJ(K)) .GT. 0.0001) ITYPE(I) = 43
      THETA(K,LCASE) = THETA(K,LCASE) + DJ(K)
  340 CONTINUE
  350 IP = 1
      RETURN
C
C   AN ERROR HAS OCCURED.    PRINT A MESSAGE AND QUIT.
C
  360 WRITE(6, 370 ) LCASE,K,N
  370 FORMAT(1H1///45X,26HFUNDAMENTAL GEOMETRIC CASE,I3/
     1        18X,50HAN INCONSISTENCY HAS BEEN FOUND IN THE HINGE INPUT,
     2        18H DATA FOR WING ROW,I3,10H, ROW TYPE,I3)
      IP = 2
      RETURN
      END


*DECK OUT1
      SUBROUTINE OUT1(LCASE)
C
C   THIS SUBROUTINE PRINTS OUT THE GEOMETRIC DATA DERIVED FROM THE
C   SECTIONAL METHOD INPUT
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,JROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON /LUKE/ TITLE(8)
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)
      COMMON/FCASE2/TST(40,10),HL(40,10),DJ(40),ACTE(40),AC(20,40),
     1             XHB(4,40),BET(4,40),IES(4,40),ICT(40),IHT(40),NCT,NHT
      COMMON/FCASE3/EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10)
      COMMON/INDATA/ARE,SPA,CRE,XM,CMA,XC,NRO,NC,ISY,IPR,JET,IGT,IHI
C
C   PRINT CASE TITLE AND GENERAL GEOMETRIC PARAMETERS
      IF(LCASE .GT. 1) GO TO 60
   10 WRITE(6, 20 ) TITLE
   20 FORMAT(1H1,39X,10(4H****)/
     1    40X,40H*  EVD  JET - WING  COMPUTER  PROGRAM  */
     2    40X,10(4H****)///20X,8A10)
      CMA = CMAC = SPA / 2.0
   30 WRITE(6, 40 ) AREA,ARE,SPAN,SPA,CREF,CRE,XMC,XM,CMAC,CMA,ARATIO,
     1             ARATIO,XCG,XC
   40 FORMAT(1H0//54X,4HUSED,11X,5HINPUT /
     1        41X,6HAREA =,2F15.6 / 41X,6HSPAN =,2F15.6 /
     2        41X,6HCREF =,2F15.6 / 42X,5HXMC =,2F15.6 /
     3        41X,6HCMAC =,2F15.6 / 39X,8HARATIO =,2F15.6 /
     4        42X,5HXCG =,2F15.6)
      WRITE(6, 50 ) NROWS,NRO,NCASES,NC,ISYMM,ISY,IPRINT,IPR,JETFLG,JET,
     1             IGTYPE,IGT,IHINGE,IHI,NWT,NJT,NMAX
   50 FORMAT(1H0/ 48X,7HNROWS =,I3,7X,I3 / 47X,8HNCASES =,I3,7X,I3 /
     1              48X,7HISYMM =,I3,7X,I3 / 47X,8HIPRINT =,I3,7X,I3 /
     2              47X,8HJETFLG =,I3,7X,I3 / 47X,8HIGTYPE =,I3,7X,I3 /
     3              47X,8HIHINGE =,I3,7X,I3 ///
     4              43X,25HNUMBER OF WING ELEMENTS =,I4 /
     5              43X,25HNUMBER OF JET ELEMENTS  =,I4 /
     6              42X,26HTOTAL NUMBER OF ELEMENTS =,I4)
   60 J = 0
      JJ = NWT
C
C   PRINT FUNDAMENTAL CASE HEADER
      WRITE(6, 70 ) LCASE
   70 FORMAT(1H1,23X,1H*,19(4H****)/
     1     24X,54H*  ELEMENT GEOMETRY DATA AND FUNDAMENTAL CASE DATA FOR,
     2     17H FUNDAMENTAL CASE,I3,3H  */24X,1H*,19(4H****))
      ILINES = 3
      DO 260 K = 1,NROWS
C
C   PRINT SECTIONAL DATA
      WRITE(6, 80 ) K,Y(K),DELTA(K),XLEAD(K),XTRAIL(K),CHORD(K),TANLE(K)
   80 FORMAT(1H0,11H*** SECTION,I3,4H ***,2X,3HY =,F10.6,2X,7HDELTA =,
     1 F10.6,2X,7HXLEAD =,F10.6,2X,8HXTRAIL =,F10.6,2X,7HCHORD =,F10.6,
     2 2X,7HTANLE =,F10.6)
C
C   PRINT CHORDWISE DATA ON WING
      NWK = NW(K)
      WRITE(6, 90 ) NWK,TST(K,LCASE),HL(K,LCASE),THS(K,LCASE)
   90 FORMAT(1H0,21HWING ELEMENTS    NW =,I3,5X, 7HTWIST =,F10.6,5X,
     1              4HHL =,F10.6,5X, 9HTHETA S =,F10.6)
      WRITE(6, 100 ) (XB(J+L),L=1,NWK)
  100 FORMAT(1H ,14X,2HXB,10F11.6 / 17X,10F11.6)
      IF(LCASE .GT. 1) GO TO 130
      WRITE(6, 110 ) (XI(J+L),L=1,NWK)
  110 FORMAT(1H ,14X, 2HXI,10F11.6/17X,10F11.6)
      WRITE(6, 120 ) (DEL(J+L),L=1,NWK)
  120 FORMAT(1H ,13X,3HDEL,10F11.6/17X,10F11.6)
  130 IF(ICT(K) .EQ. 0) GO TO 150
      ICK = ICT(K)
      WRITE(6, 140 ) (AC(L,ICK),L=1,NWK)
  140 FORMAT(1H ,10X,6HCAMBER,10F11.6/17X,10F11.6)
  150 WRITE(6, 160 ) (EPS(J+L,LCASE),L=1,NWK)
  160 FORMAT(1H ,13X,3HEPS,10F11.6/17X,10F11.6)
      WRITE(6, 170 ) (BETA(J+L,LCASE),L=1,NWK)
  170 FORMAT(1H ,12X,4HBETA,10F11.6/17X,10F11.6)
      WRITE(6, 180 ) (ITYPE(J+L),L=1,NWK)
  180 FORMAT(1H ,12X,4HTYPE,10(3X,I2,6X)/17X,10(3X,I2,6X))
      J = J + NWK
      IL = 1
      IF(NWK .GT. 9) IL = 2
      ILINES = ILINES + 4 + 4*IL
      IF(LCASE .EQ. 1) ILINES = ILINES + 2*IL
C
C   PRINT CHORDWISE DATA ON JET
      NJK = NJ(K)
      IF(NJK .GT. 0) GO TO 200
      WRITE(6, 190 )
  190 FORMAT(1H ,8X,19HTHIS ROW HAS NO JET)
      ILINES = ILINES + 1
      GO TO 230
  200 WRITE(6, 210 ) NJK,D(K),DJ(K),ACTE(K),THETA(K,LCASE)
  210 FORMAT(1H0,1X,20HJET ELEMENTS    NJ =,I3,5X,3HD =,F10.6,5X,4HDJ =,
     1    F10.6,5X,6HACTE =,F10.6,5X,7HTHETA =,F10.6)
      WRITE(6, 100 ) (XB(JJ+L),L=1,NJK)
      IF(LCASE .GT. 1) GO TO 220
      WRITE(6, 110 ) (XI(JJ+L),L=1,NJK)
      WRITE(6, 120 ) (DEL(JJ+L),L=1,NJK)
  220 WRITE(6, 170 ) (BETA(JJ+L,LCASE),L=1,NJK)
      WRITE(6, 180 ) (ITYPE(JJ+L),L=1,NJK)
      JJ = JJ + NJK
      IL = 1
      IF(NJK .EQ. 10) IL = 2
      ILINES = ILINES + 1 + 3 * IL
      IF(LCASE .EQ. 1) ILINES = ILINES + 2*IL
  230 IF(K .EQ. NROWS) GO TO 260
      NWK1 = NW(K+1)
      IL = 1
      IF(NWK1 .GT. 9) IL = 2
      NEXT = 4 + 4*IL
      IF(LCASE .EQ. 1) NEXT = NEXT + 2*IL
      NJK1 = NJ(K+1)
      IL = 1
      IF(NJK1 .EQ. 10) IL = 2
      NEXT = NEXT + 1
      IF(NJK1 .EQ. 0) GO TO 240
      NEXT = NEXT + 1 + 3*IL
      IF(LCASE .EQ. 1) NEXT = NEXT + 2*IL
  240 IF((55-ILINES) .GE. NEXT) GO TO 260
      WRITE(6, 250 )
  250 FORMAT(1H1)
      ILINES = 1
  260 CONTINUE
      RETURN
      END
```

```
*DECK INCOMP
      SUBROUTINE INCOMP(NCASES,IR)
C
C  THIS SUBROUTINE READS IN THE COMPOSITE CASE REQUIREMENTS
C  WHICH DEFINE THE FUNDAMENTAL CASES AND THEIR DEFLECTION MAGNITUDE
C  FOR SUPERPOSITION IN UP TO 24 COMBINATIONS
C
      COMMON/COMPOS/FACTOR(10,24),NCC
      DIMENSION FUNNY(10),ND(10)
C
C  INITIALIZE THE ARRAY OF FUNDAMENTAL CASE DEFLECTIONS
      DO 20 M = 1,24
      DO 10 N = 1,10
      FACTOR(N,M) = 0.00
   10 CONTINUE
   20 CONTINUE
C
C  READ THE COMPOSITE CASE DATA, CONSISTING OF FUNDAMENTAL CASE
C  DEFLECTIONS, IN DEGREES
      NCC = 0
   30 NCC = NCC + 1
      READ(5, 40 ) (ND(N),FUNNY(N),N=1,10)
   40 FORMAT(10(I2,F6.4))
      IF(EOF(5) .NE. 0) GO TO 130
C  CHECK THE VALIDITY OF THE DATA
   50 IF(ND(1) .GT. 10) GO TO 100
      IF(NCC .GT. 24) GO TO 110
      DO 90 N = 1,10
      IF(ND(N) .GT. NCASES) GO TO 70
      IF(ND(N) .LT. 1) GO TO 90
C  THE DATA IS OK.  DEFINE FACTOR.
      NDN = ND(N)
   60 FACTOR(NDN,NCC) = FUNNY(N)
      GO TO 90
   70 WRITE(6, 80 )
   80 FORMAT(1H0,22X,75HAN INCORRECT COMPOSITE CASE INPUT VALUE HAS BEEN
     1 FOUND.  IT WILL BE IGNORED.)
   90 CONTINUE
      GO TO 30
C  THE END OF THE INPUT DATA HAS BEEN REACHED
  100 NCC = NCC - 1
      IF(NCC .GT. 24) NCC = 24
      IR = 1
      RETURN
C  TOO MANY COMPOSITE CASES HAVE BEEN REQUESTED.  READ ON UNTIL AN END
C  CARD IS FOUND.
  110 WRITE(6, 120 )
  120 FORMAT(1H0,20X,47HMORE THAN 24 COMPOSITE CASES HAVE BEEN INPUT.
     1          34HSUBSEQUENT INPUTS WILL BE IGNORED.)
      GO TO 30
C
C  AN END OF FILE HAS BEEN READ.  PRINT A MESSAGE AND QUIT.
  130 WRITE(6, 140 )
  140 FORMAT(1H1/// 31X, 35HAN END OF FILE HAS BEEN READ DURING.
     1       21H COMPOSITE CASE INPUT)
      IR = 2
      RETURN
      END
```

```
*DECK BLOWIN
      SUBROUTINE BLOWIN(JETFLG,IR)
C
C  THIS SUBROUTINE READS THE SECTIONAL JET BLOWING RATES
C  CMU(K) = J / (Q * CHORD(K))
C
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)
      DIMENSION DCMU(40)
C
      IF(JETFLG .NE. 0) GO TO 30
C  READ THE CMU DATA ONLY FOR THOSE SECTIONS WHICH HAVE A JET
      READ(5, 10 ) (DCMU(K),K=1,NROWSJ)
   10 FORMAT(8F10.5)
      IF(EOF(5) .NE. 0) GO TO 60
   20 IF(DCMU(1) .LT. 800.0) GO TO 30
      IR = 2
      RETURN
C
C  REARRANGE THE DATA INTO THE PROPER SEQUENCE
   30 KP = 0
      DO 50 K = 1,NROWS
   40 CMU(K) = 0.00
      IF(NJ(K) .EQ. 0) GO TO 50
      KP = KP + 1
      CMU(K) = DCMU(KP)
   50 CONTINUE
      IR = 1
      RETURN
C
C  AN END OF FILE HAS BEEN READ.  THIS RUN IS COMPLETELY FINISHED.
   60 WRITE(6, 70 )
   70 FORMAT(1H1///41X,37HNO MORE CMU CASES HAVE BEEN REQUESTED)
      IR = 3
      RETURN
      END
```

```
*DECK BOXJ
      SUBROUTINE BOXJ(NEWMAX,IR)
C
C  THIS SUBROUTINE COMPUTES THE JET BLOWING FACTOR CMUP
C
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),KB(600),KI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)
C
C  COMPUTE THE NEW CMUP AND SAVE THE OLD VALUES AS CMUPP
   10 NEWMAX = NMAX
      ICOUNT = 0
      DO 30 K = 1,NROWS
      CMUPP(K) = CMUP(K)
      IF(NJ(K) .EQ. 0) GO TO 30
      IF(CMU(K) .LT. 0.0001) GO TO 20
      CMUP(K) = 2.00 / (CHORD(K)*CMU(K))
      GO TO 30
   20 ICOUNT = ICOUNT + 1
      CMUP(K) = 0.00
   30 CONTINUE
      WRITE(6, 40 ) (K,CMU(K),K=1,NROWS)
   40 FORMAT(1H1,40X,10(4H****)/ 41X,
     1      40H*   SECTIONAL JET BLOWING COEFFICIENTS   */41X,10(4H****)//
     2      53X,3HROW,5X,3HCMU,40(/53X,I2,F12.6))
      IF(ICOUNT .EQ. 0) GO TO 50
      IF(ICOUNT .LT. NROWSJ) GO TO 60
      NEWMAX = NWT
   50 IR = 1
      RETURN
C
C  AN ERROR HAS OCCURED.   PRINT A MESSAGE AND TRY AGAIN.
   60 WRITE(6, 70 )
   70 FORMAT(1H0, 43X,35HA ZERO VALUE OF CMU HAS BEEN INPUT..
     1        33H  THIS CMU CASE HAS BEEN IGNORED.)
      IR = 2
      RETURN
      END
```

```
*DECK STAGE2
      OVERLAY(OVLY,2,0)
      PROGRAM STAGE2
C
C  THIS PROGRAM CONTROLS THE FORMATION AND SOLUTION OF
C  THE SYSTEM OF LINEAR EQUATIONS
C
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR
C
C  FORM THE SYSTEM OF LINEAR EQUATIONS
   10 CALL OVERLAY(4HOVLY,2,1)
C
C  SOLVE THE SYSTEM OF LINEAR EQUATIONS
   20 CALL OVERLAY(4HOVLY,2,2)
      IF(IR .EQ. 2) GO TO 30
C
C  THE SOLUTION HAS NOW BEEN COMPLETED.   RETURN NORMALLY TO MAIN.
C  *** HALLELULIAH ***
      IR = 1
      GO TO 40
C
C**********************************************************************
C  A FATAL ERROR HAS OCCURED.   RETURN ABNORMALLY TO MAIN.
   30 IR = 2
   40 CONTINUE
C**********************************************************************
C
      END
```

```
*DECK STG2D
      OVERLAY(OVLY,2,1)
      PROGRAM STG2D
C
C   THIS PROGRAM CONTROLS THE CALCULATION OF ALL EVD DOWNWASH
C   INFLUENCE COEFFICIENTS AND THE FORMATION OF THE LEFT AND RIGHT SIDE
C   MATRICES.
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NCALFA,LOGIC,IR
      DIMENSION W(610)
C
C   IF THIS IS A NEW CMU CASE, AUGMENT THE EXISTING DOWNWASH MATRIX ROWS
C   ON THE JET
      ISIZE = NEWMAX
      IF(NEWCMU .EQ. 1) GO TO 10
      IF(NEWMAX .GT. NWT) CALL SHUFL2(W,ISIZE,NEWMAX)
      GO TO 30
C
C   CALCULATE THE DOWNWASH INFLUENCE COEFFICIENTS AT ALL CONTROL POINTS
C   DUE TO ALL TRIANGULAR, LEADING EDGE AND FAR-JET EVD ELEMENTS
   10 ISIZE = NMAX
      IF((LOGIC .EQ. 2) .AND. (ISYMM .GT. 0)) GO TO 30
      CALL DWNWSH(W,ISIZE)
C
C   AUGMENT THE MATRIX ROWS FOR CONTROL POINTS ON THE JET.
C   NOTE THAT THIS MUST BE DONE EVEN THOUGH CMU MAY BE 0.0,
C   IN ORDER TO PREPARE FOR FUTURE NONZERO CMU CASES.
   20 IF(NMAX .GT. NWT) CALL SHUFL1(W,ISIZE)
C
C   DEVELOP THE RIGHT SIDE COLUMN MATRIX
   30 ISIZE = NEWMAX
      DO 90 N = 1,NCASES
      LCASE = N
C
C   DEFINE THE LCASE COLUMN, NOT INCLUDING THE INFLUENCE OF ANY HINGES
   40 CALL COLUM1(LCASE)
C
C   COMPUTE THE HINGE DOWNWASH INFLUENCE FACTORS
      IF((LCASE .EQ. 1) .OR. (IHINGE .EQ. 0)) GO TO 80
      DO 50 I = 1,NEWMAX
      W(I) = 0.00
   50 CONTINUE
   60 CALL HINGE(W,ISIZE,NEWMAX,LCASE)
C
C   MODIFY THE LCASE COLUMN TO INCLUDE THE INFLUENCE OF ALL HINGES
   70 CALL COLUM2(W,ISIZE,NEWMAX,LCASE)
   80 CONTINUE
C
C   THE MATRIX DEVELOPMENT IS NOW COMPLETE.
C   PUT THE MATRIX SYSTEM IN THE PROPER FORM FOR SOLUTION.
   90 ISIZE = NEWMAX + NCASES
  100 CALL PREP(W,ISIZE,NEWMAX)
      CONTINUE
      END
```

```
*DECK DWNWSH
      SUBROUTINE DWNWSH(W,ISIZE)
C
C   THIS SUBROUTINE CALCULATES THE DOWNWASH INFLUENCE COEFFICIENT MATRIX.
C   THE MATRIX IS STORED ON THE DIRECT ACCESS SCRATCH FILE.
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XR(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      DIMENSION W(ISIZE)
C
C
C   COMPUTE ALL THE DOWNWASH COEFFICIENTS
C
      IWRITE = 0
      IF(IPRINT .LT. 0) WRITE(6, 10 )
   10 FORMAT(1H1,38X,44HWING - DUE - TO - WING - AND - JET  DOWNWASH/)
C
C   CYCLE THE DOWNWASH CONTROL POINTS ON THE WING AND JET
      DO 190 I = 1,NMAX
      IWRITE = IWRITE + 1
      KI = KK(I)
C
C   CYCLE THE VORTEX POINTS ON THE WING AND JET
      DO 150 J = 1,NMAX
C   COMPUTE THE GENERAL GEOMETRIC PARAMETERS
      KJ = KK(J)
   20 X = XI(I) + DEL(I)*CHORD(KI)/2.00 - XI(J)
      YY = Y(KI) - Y(KJ)
      IT = ITYPE(J)/10
C
C   DECIDE WHICH EVD TYPE TO USE.  ONLY THE TRIANGULAR PART OF HINGES
C   WILL BE CONSIDERED AT THIS TIME.
      GO TO ( 50 , 80 , 110 , 50 ), IT
   30 WRITE(6, 40 ) J,IT
   40 FORMAT(1H ,35X,21H** WARNING ** ELEMENT,I4,19H HAS AN ITYPE VALUE,
     1      3H OF,I3/39X,42HAN EQUIVALENT TRIANGULAR DOWNWASH WAS USED)
C
C   REGULAR TRIANGULAR EVD  (INCLUDING TRIANGULAR PART OF HINGE EVD)
   50 D1 = DEL(J-1) * CHORD(KJ)
      IN1 = IW(KJ) + NW(KJ) - 1
      IF(J .EQ. IJ(KJ))D1 = DEL(IN1) * CHORD(KJ)
      D2 = DEL(J) * CHORD(KJ)
   60 W(J) = EVD1(X,YY,D1,D2,DELTA(KJ))
C   SUPERIMPOSE THE DOWNWASH FROM THE LEFT SIDE OF THE WING IF THIS IS A
C   SYMETRIC OR ANTI-SYMETRIC CASE
      IF(ISYMM .GT. 0) GO TO 150
      YY = Y(KI) + Y(KJ)
   70 WDUMMY = EVD1(X,YY,D1,D2,DELTA(KJ))
      IF(ISYMM .LT. 0)WDUMMY = -WDUMMY
      GO TO 140
C
C   LEADING EDGE EVD
   80 D2 = DEL(J) * CHORD(KJ)
   90 W(J) = EVD2(X,YY,D2,DELTA(KJ))
C   SUPERIMPOSE DOWNWASH
      IF(ISYMM .GT. 0) GO TO 150
      YY = Y(KI) + Y(KJ)
  100 WDUMMY = EVD2(X,YY,D2,DELTA(KJ))
      IF(ISYMM .LT. 0) WDUMMY = -WDUMMY
      GO TO 140
C
C   FAR - JET EVD
  110 D1 = DEL(J-1) * CHORD(KJ)
  120 W(J) = EVD3(X,YY,D1,D(KJ),DELTA(KJ))
C   SUPERIMPOSE DOWNWASH
      IF(ISYMM .GT. 0) GO TO 150
      YY = Y(KI) + Y(KJ)
  130 WDUMMY = EVD3(X,YY,D1,D(KJ),DELTA(KJ))
      IF(ISYMM .LT. 0) WDUMMY = -WDUMMY
  140 W(J) = W(J) + WDUMMY
  150 CONTINUE
C
C   STORE THE DOWNWASH AT CONTROL POINT I ON THE DIRECT ACCESS UNIT
C
  160 CALL WRITMS(1,W,ISIZE,IWRITE)
      IF(IPRINT .GE. 0) GO TO 190
      IF(I .EQ. NWT+1) WRITE(6, 170 )
  170 FORMAT(1H1,38X,43HJET - DUE - TO - WING - AND - JET  DOWNWASH/)
      WRITE(6, 180 ) I,W
  180 FORMAT(1H0,55X,10HMATRIX ROW,I4,60(/1X,10F13.5))
  190 CONTINUE
      RETURN
C
C   THE DIRECT ACCESS UNIT NOW CONTAINS THE FOLLOWING -
C       WING-DUE-TO-WING-AND-JET DOWNWASH COEFFICIENTS  (NWT RECORDS)
C       JET -DUE-TO-WING-AND-JET DOWNWASH COEFFICIENTS  (NJT RECORDS)
      END
```

```
*DECK EVD1
      FUNCTION EVD1(X,Y,D1,D2,DELTA)
C
C   THIS FUNCTION CALCULATES THE DOWNWASH AT ANY POINT X,Y
C   DUE TO A REGULAR TRIANGULAR EVD ELEMENT WITH UNIT PEAK VORTICITY,
C   LOCATED AT THE ORIGIN 0,0
C
      R(A,B) = SQRT(A*A + B*B)
C
C   CALCULATE THE BASIC GEOMETRICAL PARAMETERS
      IF(Y .LT. 0.0) Y = -Y
      YMD = Y - DELTA
      YPD = Y + DELTA
      PART1 = (D1 + D2) * (1.0/YMD - 1.0/YPD)
      IF(X/(0.50*(D1+D2)) .GT. 100.0) GO TO 90
   10 XPD = X + D1
      XMD = X - D2
   20 R0P = R(X,YMD)
      R1P = R(XPD,YMD)
      R2P = R(XMD,YMD)
   30 R0PP = R(X,YPD)
      R1PP = R(XPD,YPD)
      R2PP = R(XMD,YPD)
C
C   CALCULATE THE DOWNWASH
   40 PART2 = (XPD/D1) * ((R1P-R0P)/YMD - (R1PP-R0PP)/YPD)
      PART3 = (XMD/D2) * ((R2P-R0P)/YMD - (R2PP-R0PP)/YPD)
      YRATIO = (YPD+R0PP) / (YMD+R0P)
   50 PART4 = (XPD/D1) * ALOG(((YMD+R1P)/(YPD+R1PP)) * YRATIO)
      PART5 = (XMD/D2) * ALOG(((YMD+R2P)/(YPD+R2PP)) * YRATIO)
   60 PART6 = YMD * ALOG((XPD+R1P)/(X+R0P))
      PART7 = YPD * ALOG((XPD+R1PP)/(X+R0PP))
   70 PART8 = YMD * ALOG((XMD+R2P)/(X+R0P))
      PART9 = YPD * ALOG((XMD+R2PP)/(X+R0PP))
   80 EVD1 = -(PART1 + (PART2 + PART3) - 2.0*(PART4 + PART5)
     1      -((PART6 - PART7)/D1 + (PART8 - PART9)/D2)) / 25.13274
      RETURN
   90 EVD1 = -PART1 / 12.56673
  100 RETURN
      END
```

```
*DECK EVD2
      FUNCTION EVD2(X,Y,DEL,DELTA)
C
C   THIS FUNCTION CALCULATES THE DOWNWASH AT ANY POINT X,Y
C   DUE TO A LEADING EDGE EVD ELEMENT WITH UNIT AVERAGE VORTICITY,
C   LOCATED AT THE ORIGIN 0,0
C
      DIMENSION SI(9),FACTOR(9)
C
      R(A,B) = SQRT(A*A + B*B)
      G(A) = 1.00/SQRT(A) - A
      S(A) = ABS(A) / A
C
      DATA SI/-0.9681602,-0.8360311,-0.6133714,-0.3242534,0.0,
     1        0.3242534, 0.6133714, 0.8360311, 0.9681602/
      DATA FACTOR/0.0812744,0.1806482,0.2606107,0.3123471,0.3302394,
     1        0.3123471,0.2606107,0.1806482,0.0812744/
C
C   CALCULATE THE BASIC GEOMETRICAL PARAMETERS
   10 XB = X / DEL
      YB = Y / DEL
      DB = DELTA / DEL
      YPD = YB + DB
      YMD = YB - DB
   20 IF(XB .GT. 100.0) GO TO 280
      XM1 = XB - 1.00
   30 RCP = R(XB,YMD)
   40 RCPP = R(XB,YPD)
C
C   CALCULATE RK(XB)
      IF(ABS(XB) .LT. 1.0E-04) GO TO 100
      IF(ABS(XM1) .LT. 1.0E-06) GO TO 80
   50 PART = ALOG(ABS(XM1/XB))
      PART1 = XB * PART + 1.00
      IF(XB .GT. 0.00) GO TO 70
   60 SQX = SQRT(-XB)
      RK = -2.00 / SQX * ATAN(1.00/SQX) + PART1
      GO TO 90
   70 SQX = SQRT(XB)
      RK = -ALOG(ABS((1.00-SQX) /(1.00+SQX))) / SQX + PART1
      GO TO 90
   80 RK = 2.386294
C
C   CALCULATE P(XB)
   90 PART2 = RCP/YMD - RCPP/YPD
      P = PART2 - RK
      GO TO 110
  100 P = 0.00
C
C   CALCULATE F(XB) BY GAUSSIAN INTEGRATION.
  110 IF((XB .GT. 0.0) .AND. (XB .LT. 1.00)) GO TO 150
C   XB IS NOT WITHIN THE X DIMENSIONS OF THE ELEMENT.
      F = 0.00
      DO 140 N = 1,9
  120 SB = (SI(N)+1.00) / 2.00
      XMS = XB - SB
      GS = G(SB)
  130 PART3 = (GS*(R(XMS,YMD)-RCP))/YMD - (GS*(R(XMS,YPD)-RCPP))/YPD
      F = F + FACTOR(N) * PART3 / XMS
  140 CONTINUE
      F = 0.50 * F
      GO TO 270
C
C   XB IS WITHIN THE X DIMENSIONS OF THE ELEMENT.   CALCULATE F0.
  150 F0 = 0.00
      GPX = 0.00
      GPPX = 0.00
      IF(XB .LT. 1.0E-04) GO TO 170
      GX = G(XB)
      GPX = GX * (ABS(YMD) - RCP)
      GPPX = GX * (ABS(YPD) - RCPP)
      IF(XM1 .GT. -1.0E-06) GO TO 170
  160 F0 = -(GPX/YMD - GPPX/YPD) * PART
C   CALCULATE F1 BY GAUSSIAN INTEGRATION.
  170 F1 = 0.00
      DO 250 N = 1,9
  180 SB = (SI(N)+1.00) / 2.00
      XMS = XB - SB
  190 IF(ABS(XMS) .LT. 1.0E-04) GO TO 220
      GS = G(SB)
  200 PART4 = (GS*(R(XMS,YMD)-RCP) - GPX) / XMS
  210 PART5 = (GS*(R(XMS,YPD)-RCPP) - GPPX) / XMS
      PART6 = PART4/YMD - PART5/YPD
      GO TO 240
  220 PART4 = S(YMD) - S(YPD) - PART2
      PART5 = 1.00 + 0.50 / (SQRT(XB))**3
  230 PART6 = PART4 * PART5
  240 F1 = F1 + FACTOR(N) * PART6
  250 CONTINUE
  260 F = F0 + 0.50 * F1
C
C   CALCULATE THE DOWNWASH INFLUENCE COEFFICIENT.
  270 EVD2 = -(1.50 * (1.00/YMD - 1.00/YPD) + P + F) / 18.84956
      RETURN
  280 EVD2 = -(1.00/YMD - 1.00/YPD) / 6.283185
      RETURN
      END
```

```
*DECK EVD3
      FUNCTION EVD3(X,Y,DEL,D,DELTA)
C
C   THIS FUNCTION CALCULATES THE DOWNWASH AT ANY POINT X,Y DUE TO A
C   FAR-JET EVD ELEMENT WITH UNIT PEAK VORTICITY, LOCATED AT THE ORIGIN
C
      R(A,B) = SQRT(A*A + B*B)
C
C   CALCULATE THE BASIC GEOMETRICAL PARAMETERS
      IF(Y .LT. 0.0) Y = -Y
      YMD = Y - DELTA
      YPD = Y + DELTA
   10 XPD = X + D
   20 PART1 = (DEL/2.00 + D) * (1.00/YMD - 1.00/YPD)
C   CHECK ON INFINITY
      IF((XPD/YMD)**2 .GT. 1.0E06) GO TO 160
      XPD1 = X + DEL
      XD = X/DEL
      RCP = R(X,YMD)
      RCPP = R(X,YPD)
   30 R1P = R(XPD1,YMD)
      R1PP = R(XPD1,YPD)
      RDP = R(XPD,YMD)
      RDPP = R(XPD,YPD)
C
C   CALCULATE F0
   40 PART2 = RDP/YMD - RCPP/YPD
      PART3 = 0.50 * (XD+1.00) * ((R1P-RDP)/YMD - (R1PP-RDPP)/YPD)
   50 PART4 = ALOG(ABS(((YMD+R1P)/(YPD+R1PP)) * ((YPD+RDPP)/(YMD+RDP))))
      PART5 = YMD/DEL * ALOG((XPD1+R1P)/(X+RCP))
   60 PART6 = YPD/DEL * ALOG((XPD1+R1PP)/(X+RCPP))
   70 F0 = PART1 - 0.50*PART2 + PART3 - (XD+1.00)*PART4
     1       - 0.50*(PART5-PART6)
C
C   CALCULATE F1
      IF(ABS(XPD/D) .LT. 1.0E-02) GO TO 130
C   X IS NOT NEAR -D
      Q = D/XPD
   80 PART1 = -D * Q * (1.00/YMD - 1.00/YPD)
      PART2 = Q * PART2
   90 PART3 = Q * Q * ALOG(ABS((YMD+RCP)/(YPD+RCPP)))
  100 PART4 = YMD/RDP*ALOG(ABS((-XPD+RDP*((RCP+RDP)/D))/(RDP-XPD)))
  110 PART5 = YPD/RDPP*ALOG(ABS((-XPD+RDPP*((RCPP+RDPP)/D))/(RDPP-XPD)))
  120 F1 = PART1 + PART2 - PART3 + Q * Q * (PART4 - PART5)
      GO TO 150
C   X IS NEAR -D
  130 X = -D
      RCP = R(X,YMD)
      RCPP = R(X,YPD)
      PART2 = RDP/YMD - RCPP/YPD
      PART3 = (X/YMD)**2 * ALOG(ABS((YMD+RCP)/X))
      PART4 = (X/YPD)**2 * ALOG(ABS((YPD+RCPP)/X))
  140 F1 = -0.50*PART2 - 0.50*(PART3 - PART4)
C   CALCULATE THE DOWNWASH INFLUENCE COEFFICIENT
C
  150 EVD3 = -(F0 + F1) / 12.56637
      RETURN
  160 EVD3 = -PART1 / 6.283185
      RETURN
      END
```

```
*DECK EVD4
      FUNCTION EVD4(X,Y,D1,D2,DELTA)
C
C  THIS FUNCTION CALCULATES THE DOWNWASH AT ANY POINT X,Y
C  DUE TO A HINGE EVD ELEMENT WITH ONE RADIAN TURNING ANGLE
C  LOCATED AT THE ORIGIN 0,0
C
      DIMENSION SI(9),FACTOR(9)
      R(A,B) = SQRT(A*A + B*B)
      CHANGE(A,B,C) = 0.50 * (C * (B-A) + (A+B))
      S(A) = ABS(A) / A
      SO(T) = 0.50 * (-D1L/D1*(1.00-S(T)) + D2L/D2*(1.00+S(T)))
      G(A) = ALOG(ABS(A)) - SO(A) * A
      DATA SI/-0.9681602,-0.8360311,-0.6133714,-0.3242534,0.0,
     1        0.3242534, 0.6133714, 0.8360311, 0.9681602/
      DATA FACTOR/0.0812744,0.1806482,0.2606107,0.3123471,0.3302394,
     1        0.3123471,0.2606107,0.1806482,0.0812744/
C
C  CALCULATE THE BASIC GEOMETRICAL PARAMETERS.
      DB = 0.50 * (D1 + D2)
   10 XB = X / DB
      XMD = X - D2
      XPD = X + D1
   20 YMD = Y - DELTA
      YPD = Y + DELTA
   30 ROP = R(X,YMD)
      ROPP = R(X,YPD)
      AX = ABS(X)
      AXB = ABS(XB)
      D1L = ALOG(D1)
      D2L = ALOG(D2)
      XD1 = X/D1
      XD2 = X/D2
   40 PART5 = 1.00/YMD - 1.00/YPD
      PART6 = -(D1+D2) + 0.50 * (D1*D1L + D2*D2L)
C
C  CALCULATE RK(X)
      IF(AXB .LT. 7.5) GO TO 70
      N = 0
      RK = 0.00
      D1X = D1/X
      D2X = D2/X
      DO 60 N1 = 1,100,2
      RKN = 0.000
      DO 50 N2 = 1,2
      N = N + 1
      RN = N
      RNN = N * (N+1)
      RKN = RKN + (-1.0)**(N+1) * (D1L/RNN - 1.0/(RN*RN)) * D1X**N
     1      +(D2L/RNN - 1.0/(RN*RN)) * D2X**N
   50 CONTINUE
      RK = RK + RKN
      IF(ABS(RKN/RK) .LT. 1.0E-07) GO TO 190
   60 CONTINUE
   70 IF(AXB .LT. 1.0E-04) GO TO 200
      SX = S(X)
      RK1 = 0.00
      RK2 = 0.00
      RK3 = 0.00
      RK4 = 0.00
      RK1P = ABS(XD1+1.00)
   80 IF(RK1P .LT. 1.0E-06) GO TO 90
      RK1 = (ALOG(ABS(XD1)) + (XD1+1.00) * D1L) * ALOG(ABS(XPD/X))
   90 RK2P = ABS(XD2-1.00)
      IF(RK2P .LT. 1.0E-06) GO TO 100
      RK2 = -(ALOG(ABS(XD2)) - (XD2-1.00) * D2L) * ALOG(ABS(XMD/X))
  100 IF(ABS(D1/AX-1.00) .LT. 1.0E-04) GO TO 140
C  CALCULATE RK3 BY GAUSSIAN INTEGRATION.
      AL = 1.00
      BL = D1/AX
      DO 130 N = 1,9
  110 T = CHANGE(AL,BL,SI(N))
  120 RK3 = RK3 + FACTOR(N) * (ALOG(T) / (SX+T))
  130 CONTINUE
      RK3 = 0.50 * (BL-AL) * RK3
C  CALCULATE RK4 BY GAUSSIAN INTEGRATION.
  140 IF(ABS(D2/AX-1.00) .LT. 1.0E-04) GO TO 180
      AL = 1.00
      BL = D2/AX
      DO 170 N = 1,9
  150 T = CHANGE(AL,BL,SI(N))
  160 RK4 = RK4 + FACTOR(N) * (ALOG(T) / (SX-T))
  170 CONTINUE
      RK4 = 0.50 * (BL-AL) * RK4
  180 RK = -2.467401 * SX - (D1L-D2L) + (RK1 + RK2) + (RK3 + RK4)
  190 P = (ROP/YMD - ROPP/YPD) * RK
      GO TO 210
  200 P = 0.00
C
C  CALCULATE F(X) BY GAUSSIAN INTEGRATION.
  210 IF((X .GT. -D1) .AND. (X .LT. D2)) GO TO 290
C  X IS NOT WITHIN THE DIMENSIONS OF THE ELEMENT.
C  LEFT SIDE INTEGRAL.
      FL = 0.00
      AL = -D1
      BL = 0.00
      DO 240 N = 1,9
  220 SY = CHANGE(AL,BL,SI(N))
      XMS = X - SY
      GS = G(SY)
  230 PART1 = GS * (R(XMS,YMD)-ROP) / YMD - GS * (R(XMS,YPD)-ROPP) / YPD
      FL = FL + FACTOR(N) * PART1 / XMS
  240 CONTINUE
      FL = 0.50 * (BL-AL) * FL
C  RIGHT SIDE INTEGRAL
      FR = 0.00
      AL = 0.00
      BL = D2
      DO 270 N = 1,9
  250 SY = CHANGE(AL,BL,SI(N))
      XMS = X - SY
      GS = G(SY)
  260 PART1 = GS * (R(XMS,YMD)-ROP) / YMD - GS * (R(XMS,YPD)-ROPP) / YPD
      FR = FR + FACTOR(N) * PART1 / XMS
  270 CONTINUE
      FR = 0.50 * (BL-AL) * FR
  280 F = FL + FR
      GO TO 460
C
C  X IS WITHIN THE DIMENSIONS OF THE ELEMENT
C  CALCULATE F0
  290 F0 = 0.00
      GPX = 0.00
      GPPX = 0.00
      IF(AXB .LT. 1.0E-04) GO TO 310
      SOX = SO(X) - 1.00/X
      GX = G(X)
      GPX = GX * (ABS(YMD)-ROP)
      GPPX = GX * (ABS(YPD)-ROPP)
      IF(((1.00-XD2) .LT. 1.0E-06) .OR. ((XD1+1.00) .LT. 1.0E-06))
     1        GO TO 310
  300 F0 = -(GPX/YMD - GPPX/YPD) * ALOG(ABS(XMD/XPD))
C
C  CALCULATE F1
C  LEFT SIDE INTEGRAL
  310 FL = 0.00
      AL = -D1
      BL = 0.00
      DO 370 N = 1,9
  320 SY = CHANGE(AL,BL,SI(N))
      XMS = X - SY
      IF(ABS(XMS/DB) .LT. 1.0E-04) GO TO 350
      GS = G(SY)
  330 PART2 = (GS * (R(XMS,YMD)-ROP) - GPX) / XMS
  340 PART3 = (GS * (R(XMS,YPD)-ROPP) - GPPX) / XMS
      PART4 = PART2/YMD - PART3/YPD
      GO TO 360
  350 PART2 = ROP/YMD - ROPP/YPD
      PART4 = (S(YMD) - S(YPD) - PART2) * SOX
  360 FL = FL + FACTOR(N) * PART4
  370 CONTINUE
      FL = 0.50 * (BL-AL) * FL
C
C  RIGHT SIDE INTEGRAL
  380 FR = 0.00
      AL = 0.00
      BL = D2
      DO 440 N = 1,9
  390 SY = CHANGE(AL,BL,SI(N))
      XMS = X - SY
      IF(ABS(XMS/DB) .LT. 1.0E-04) GO TO 420
      GS = G(SY)
  400 PART2 = (GS * (R(XMS,YMD)-ROP) - GPX) / XMS
  410 PART3 = (GS * (R(XMS,YPD)-ROPP) - GPPX) / XMS
      PART4 = PART2/YMD - PART3/YPD
      GO TO 430
  420 PART2 = ROP/YMD - ROPP/YPD
      PART4 = (S(YMD) - S(YPD) - PART2) * SOX
  430 FR = FR + FACTOR(N) * PART4
  440 CONTINUE
      FR = 0.50 * (BL-AL) * FR
  450 F = FL + FR + F0
C
C  CALCULATE THE DOWNWASH INFLUENCE COEFFICIENT
  460 EVD4 = (PART5 * PART6 + P + F) / 19.739202
      RETURN
      END
```

```
*DECK SHUFL1
      SUBROUTINE SHUFL1(W,ISIZE)
C
C   THIS SUBROUTINE READS THE PORTION OF THE DOWNWASH MATRIX WHICH
C   CONTAINS THE DOWNWASH DUE TO THE JET, AUGMENTS IT ACCORDING TO
C   THE CURRENT CMU VALUES, AND WRITES IT BACK ON UNIT 1
C   BEHIND THE DOWNWASH MATRIX
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)
      DIMENSION A(600),AIM1(600),W(ISIZE)
C
      NWT1 = NWT + 1
      IF(IPRINT .LT. 0) WRITE(6, 10 )
   10 FORMAT(1H1,42X,35HAUGMENTED PORTION OF SOLUTION MATRIX/)
C
C   PREPARE THE SOLUTION MATRIX FOR ROWS ON THE JET
C
C       I IS THE COUNTER FOR IDENTIFYING ELEMENTS ON THE JET
C   IREAD IS THE COUNTER FOR DOWNWASH ROWS ON THE JET STORED ON UNIT 1
C  IWRITE IS THE COUNTER FOR AUGMENTED ROWS TO BE WRITTEN ON UNIT 1
   20 IREAD = NWT
      IWRITE = NMAX
      C1 = 0.125000
      C3 = 0.375000
      DO 150 I = NWT1,NMAX
   30 IM1 = I - 1
      IP1 = I + 1
      IREAD = IREAD + 1
      IWRITE = IWRITE + 1
      K = KK(I)
C
C   READ THE ITH ROW OF THE DOWNWASH MATRIX  (IREADTH RECORD)
   40 CALL READMS(1,W,ISIZE,IREAD)
C
C   SAVE THE EXISTING ROW OF SIMPLE DOWNWASH COEFFICIENTS
      DO 50 J = 1,NMAX
      A(J) = W(J)
   50 CONTINUE
C
C   SUBTRACT THE PREVIOUS ROW FROM THE PRESENT ROW IF THE DOWNWASH POINT
C   IS NOT ON A LEADING JET ELEMENT
      PROD1 = CMUP(K) * DEL(I) * CHORD(K)
      PROD2 = CMUP(K) * DEL(IM1) * CHORD(K)
   60 IF(I .EQ. IJ(K)) GO TO 90
      DO 70 J = 1,NMAX
      W(J) = W(J) - AIM1(J)
   70 CONTINUE
C
C   MODIFY THE TWO OR THREE SPECIAL ELEMENTS FURTHER
      IF(I .EQ. (IJ(K)+NJ(K)-1)) GO TO 100
C   DOWNWASH CONTROL POINT IS ON A REGULAR JET ELEMENT
   80 W(IM1) = W(IM1) + C1 * PROD2
      W(I) = W(I) + C3 * (PROD1 + PROD2)
      W(IP1) = W(IP1) + C1 * PROD1
      GO TO 110
C   DOWNWASH CONTROL POINT IS ON A LEADING JET ELEMENT
   90 W(I) = W(I) + C3 * PROD1
      W(IP1) = W(IP1) + C1 * PROD1
      GO TO 110
C   DOWNWASH CONTROL POINT IS ON A TRAILING JET ELEMENT
  100 W(IM1) = W(IM1) + C1 * PROD2
      W(I) = W(I) + C3 * PROD2 + CMUP(K) * D(K)
C
C   STORE THE AUGMENTED ITH ROW ON UNIT 1  (IWRITETH RECORD)
  110 CALL WRITMS(1,W,ISIZE,IWRITE)
C   PRINT THE AUGMENTED PORTION OF THE MATRIX
      IF(IPRINT .LT. 0) WRITE(6, 120 ) I,W
  120 FORMAT(1H0,55X,10HMATRIX ROW,I4,60(/1X,10E13.5))
C
C   SAVE THE ITH ROW FOR USE AS THE I-1 ROW ON THE NEXT PASS
  130 DO 140 J = 1,NMAX
      AIM1(J) = A(J)
  140 CONTINUE
  150 CONTINUE
      RETURN
C
C   DIRECT ACCESS UNIT 1 NOW CONTAINS THE FOLLOWING -
C      WING-DUE-TO-WING-AND-JET DOWNWASH COEFFICIENTS (NWT RECORDS)
C       JET-DUE-TO-WING-AND-JET DOWNWASH COEFFICIENTS (NJT RECORDS)
C       JET-DUE-TO-WING-AND-JET AUGMENTED DOWNWASH COEFFICIENTS (NJT RECS)
      END
```

```
*DECK SHUFL2
      SUBROUTINE SHUFL2(W,ISIZE,NEWMAX)
C
C   THIS SUBROUTINE READS EACH MATRIX ROW CORRESPONDING TO A DOWNWASH
C   CONTROL POINT ON THE JET, MODIFIES IT ACCORDING TO THE NEW VALUES
C   OF CMU, AND RESTORES IT IN ITS ORIGINAL PLACE
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)
      DIMENSION W(ISIZE)
C
      IF(IPRINT .LT. 0) WRITE(6, 10 )
   10 FORMAT(1H1,42X,36HAUGMENTED PORTION OF SOLUTION MATRIX)
C
C   CYCLE THE AUGMENTED MATRIX ROWS
      IREAD = NMAX
      NWT1 = NWT + 1
      DO 100 I = NWT1,NEWMAX
      IREAD = IREAD + 1
   20 IM1 = I - 1
      IP1 = I + 1
      K = KK(I)
   30 CMUDIF = (CMUP(K) - CMUPP(K)) * CHORD(K)
C
C   READ THE ITH AUGMENTED MATRIX ROW
   40 CALL READMS(1,W,ISIZE,IREAD)
C
C   MODIFY THE TWO OR THREE SPECIAL ELEMENTS ACCORDING TO THE NEW CMU
      IF(I .EQ. IJ(K)) GO TO 60
      IF(I .EQ. (IJ(K)+NJ(K)-1)) GO TO 70
C   DOWNWASH CONTROL POINT IS ON A REGULAR JET ELEMENT
   50 W(IM1) = W(IM1) + 0.1250 * DEL(IM1) * CMUDIF
      W(I) = W(I) + 0.3750 * (DEL(IM1)+DEL(I)) * CMUDIF
      W(IP1) = W(IP1) + 0.1250 * DEL(I) * CMUDIF
      GO TO 80
C   DOWNWASH CONTROL POINT IS ON A LEADING JET ELEMENT
   60 W(I) = W(I) + 0.3750 * DEL(I) * CMUDIF
      W(IP1) = W(IP1) + 0.1250 * DEL(I) * CMUDIF
      GO TO 80
C   DOWNWASH CONTROL POINT IS ON A TRAILING JET ELEMENT
   70 W(IM1) = W(IM1) + 0.1250 * DEL(IM1) * CMUDIF
      W(I) = W(I) + (0.3750*DEL(IM1) + D(K)/CHORD(K)) * CMUDIF
C
C   WRITE THE REVISED ITH ROW ON UNIT 1
   80 CALL WRITMS(1,W,ISIZE,IREAD)
      IF(IPRINT .LT. 0) WRITE(6, 90 ) I,W
   90 FORMAT(1H0,55X,10HMATRIX ROW,I4,60(/1X,10E13.5))
  100 CONTINUE
      RETURN
      END
```

```
*DECK COLUM1
      SUBROUTINE COLUM1(LCASE)
C
C   THIS SUBROUTINE SETS UP THE RIGHT SIDE COLUMN MATRIX WITHOUT
C   CONSIDERATION OF ANY HINGE DOWNWASH INFLUENCE
C
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/FCASE3/EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10)
      COMMON/SOLV1/B(600,10)
C
C   DEFINE THE ELEMENTS ON THE WING
      DO 10 I = 1,NWT
      B(I,LCASE) = EPS(I,LCASE) / 57.295779
   10 CONTINUE
C
C   DEFINE THE ELEMENTS ON THE JET
      I = NWT
      DO 40 K = 1,NROWS
      NJK = NJ(K)
      IF(NJK .EQ. 0) GO TO 40
C   FIRST JET ELEMENT
      I = I + 1
      KKI = KK(I)
   20 B(I,LCASE) = THETA(KKI,LCASE) / 57.295779
C   REMAINING JET ELEMENTS
      DO 30 L = 2,NJK
      I = I + 1
      B(I,LCASE) = 0.00
   30 CONTINUE
   40 CONTINUE
      RETURN
      END
```

```
*DECK COLUM2
      SUBROUTINE COLUM2(H,ISIZE,NEWMAX,LCASE)
C
C  THIS SUBROUTINE ADDS THE APPROPRIATE HINGE DOWNWASH INFLUENCE
C  TO THE RIGHT SIDE COLUMN MATRIX
C
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/ECASE3/EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10)
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)
      COMMON/SOLV1/B(600,10)
      DIMENSION H(ISIZE)
C
C  DEFINE THE ELEMENTS ON THE WING
      DO 10 I = 1,NEWMAX
      B(I,LCASE) = B(I,LCASE) - H(I)
   10 CONTINUE
C
C  DEFINE THE ELEMENTS ON THE JET
      IF(NEWMAX .EQ. NWT) RETURN
      I = NWT
      DO 90  K = 1,NROWS
   20 NJK = NJ(K)
      IF(NJK .EQ. 0) GO TO 90
C  COMPUTE THE CMU INFLUENCE FACTORS H1 AND H2
      I = I + 1
      H1 = 0.00
      H2 = 0.00
      BTA = BETA(I,LCASE)
      IF(ABS(BTA) .LT. 0.0001) GO TO 50
      BTA = BTA / 57.295779
   30 D2 = DEL(I) * CHORD(K)
      DL = ALOG(D2)
      PROD = -CMUP(K) * D2 * BTA / 3.1415927
   40 H1 = PROD * (1.6931472 - 0.750 * DL)
      H2 = PROD * (0.3068528 - 0.250 * DL)
C  FIRST POINT ON THE JET
   50 B(I,LCASE) = B(I,LCASE) + H1
C  SECOND POINT ON THE JET
      I = I + 1
   60 B(I,LCASE) = B(I,LCASE) + H(I-1) + H2
C  REMAINING POINTS ON THE JET
      IF(NJK .LT. 3) GO TO 90
      DO 80 L = 3,NJK
      I = I + 1
   70 B(I,LCASE) = B(I,LCASE) + H(I-1)
   80 CONTINUE
   90 CONTINUE
      RETURN
      END


*DECK HINGE
      SUBROUTINE HINGE(H,ISIZE,NEWMAX,LCASE)
C
C  THIS SUBROUTINE CALCULATES THE DOWNWASH INFLUENCE COEFFICIENTS
C  AT EACH DOWNWASH CONTROL POINT DUE TO ALL DEFLECTED HINGE ELEMENTS.
C  FOR EACH CONTROL POINT THE INFLUENCE COEFFICIENTS ARE MULTIPLIED BY
C  THEIR RESPECTIVE DEFLECTION ANGLE AND SUMMED UP TO OBTAIN THE
C  COMPLETE HINGE-INDUCED DOWNWASH.
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/ECASE3/EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10)
      DIMENSION H(ISIZE)
C
      IF(LCASE .GT. 2) GO TO 20
      IF(IPRINT .LT. 0) WRITE(6, 10 )
   10 FORMAT(1H1)
      ILINES = 1
C
C  CYCLE THE DOWNWASH CONTROL POINTS ON THE WING AND JET
   20 DO 110 I = 1,NEWMAX
      KI = KK(I)
C
C  CYCLE THE VORTEX POINTS ON THE WING AND JET
      DO 100 J = 1,NEWMAX
C  CHECK WHETHER THERE IS A DEFLECTED HINGE AT ELEMENT J
   30 B = BETA(J,LCASE)
      IF(ABS(B) .LT. 0.0001) GO TO 100
      B = B / 57.295779
C  COMPUTE THE GEOMETRIC PARAMETERS
      KJ = KK(J)
   40 X = XI(I) + DEL(I)*CHORD(KI)/2.00 - XI(J)
      YY = Y(KI) - Y(KJ)
   50 D2 = DEL(J) * CHORD(KJ)
      D1 = DEL(J-1) * CHORD(KJ)
      IN1 = IW(KJ) + NW(KJ) - 1
   60 IF(ITYPE(J) .EQ. 43) D1 = DEL(IN1) * CHORD(KJ)
C  COMPUTE AND SUM UP THE INFLUENCE OF ELEMENT J
   70 H(I) = H(I) + EVD4(X,YY,D1,D2,DELTA(KJ)) * B
C  SUPERIMPOSE DOWNWASH FOR SYMETRIC OR ANTI-SYMETRIC GEOMETRY
      IF(ISYMM .GT. 0) GO TO 100
      YY = Y(KI) + Y(KJ)
   80 HDUMMY = EVD4(X,YY,D1,D2,DELTA(KJ))
      IF(ISYMM .LT. 0) HDUMMY = - HDUMMY
   90 H(I) = H(I) + HDUMMY * B
  100 CONTINUE
  110 CONTINUE
C
C  PRINT OUT THE DOWNWASH IF REQUIRED
      IF(IPRINT .GE. 2) RETURN
      NEXT = NEWMAX/10 + 3
      IF(((ILINES+NEXT) .LT. 56) .OR. (ILINES .EQ. 1)) GO TO 120
      WRITE(6, 10 )
      ILINES = 1
  120 IF(IPRINT .LT. 2) WRITE(6, 130 ) LCASE,H
  130 FORMAT(1H0,35X,44HHINGE INFLUENCE COEFFICIENTS FOR FUNDAMENTAL,
     1       5H CASE, I3,60(/1X,10F13.5))
      ILINES = ILINES + NEXT
      RETURN
      END


*DECK PREP
      SUBROUTINE PREP(TRANS,ISIZE,NEWMAX)
C
C  THIS SUBROUTINE PREPARES THE FINAL MATRIX FOR SOLUTION BY
C  CONCATINATING IT WITH THE RIGHT SIDE MATRIX AND STORING IT
C  ON SCRATCH UNIT 2 FOR INPUT TO MATRIX.
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/SOLV1/B(600,10)
      DIMENSION TRANS(ISIZE)
C
      IREAD = 0
      REWIND 2
C
C  READ THE MATRIX COEFFICIENTS
      DO 40  I = 1,NEWMAX
      IF(I .EQ. NWT+1) IREAD = NEWMAX
      IREAD = IREAD + 1
      CALL READMS(1,TRANS,ISIZE,IREAD)
C
C  PULL OUT THE RIGHT SIDE MATRIX COEFFICIENTS AND CONCATINATE THEM
   10 DO 20 N = 1,NCASES
      TRANS(NEWMAX+N) = B(I,N)
   20 CONTINUE
C  THE MATRIX ROW HAS NOW BEEN ASSEMBLED AND FILLS THE TRANS ARRAY.
C  WRITE THE CONCATINATED ROW ON ON UNIT 2 FOR INPUT TO MATRIX.
   30 WRITE(2) TRANS
   40 CONTINUE
C  THE SYSTEM OF LINEAR EQUATIONS IS NOW READY FOR SOLUTION
      RETURN
      END


*DECK STG2S
      OVERLAY(OVLY,2,2)
      PROGRAM STG2S
C
C  THIS PROGRAM CONTROLS SOLUTION OF THE MATRIX SYSTEM
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR
      COMMON/SOLV1/GAMMA(600,10)
      COMMON/SOLV2/WKAREA(10000)
      DIMENSION W(600),SUMMER(600)
C
      IF(IPRINT .LT. 0) WRITE(6, 10 )
   10 FORMAT(1H1,53X,14HGAMMA SOLUTION)
C
C  SOLVE THE MATRIX FOR GAMMA USING MATRIX
      NSIZE = 10000
      NIN = 2
      NSCR1 = 3
      NSCR2 = 4
   20 CALL MATRIX(NEWMAX,NCASES,NSIZE,NIN,NSCR1,NSCR2,IR)
      IF(IR .EQ. 2) GO TO 90
C  MATRIX HAS STORED THE SOLUTION IN THE FIRST STORAGE LOCATIONS OF THE
C  WKAREA ARRAY.  TRANSFER THIS DATA INTO THE GAMMA ARRAY.
      ISUM = 0
   30 DO 70 N = 1,NCASES
      DO 40 J = 1,NEWMAX
      GAMMA(J,N) = WKAREA(ISUM+J)
   40 CONTINUE
      IF(IPRINT .LT. 0) WRITE(6, 50 ) N,(GAMMA(J,N),J=1,NEWMAX)
   50 FORMAT(1H0,50X,16HFUNDAMENTAL CASE,I4,60(/1X,10F13.5))
   60 ISUM = ISUM + NEWMAX
   70 CONTINUE
C
C  IF REQUIRED, CHECK THE SOLUTION BY BACK SUBSTITUTION
   80 IF(IPRINT .LT. 0) CALL BAKSUB(W,SUMMER,NEWMAX)
      IR = 1
      GO TO 110
C
C  PRINT THE MATRIX ERROR MESSAGE
   90 WRITE(6, 100 )
  100 FORMAT(1H1,40X,40HMATRIX DOES NOT HAVE ENOUGH CORE TO WORK/
     1       45X,29HTHIS CASE HAS BEEN TERMINATED)
      IR = 2
  110 CONTINUE
      END
```

```
*DECK MATRIX
      SUBROUTINE MATRIX(ND, MD, KD, NI, MM, NO, IR)
C
C               D I R E C T   M A T R I X   S O L U T I O N
C
      COMMON/SOLV2/A(10000)
      LOGICAL  LAST
C
      N = ND
      M = MD
      KORE = KD
      NPM = N + M
      IF (MAX0(3 * NPM, M * N) .LT. KORE) GO TO 20
      IR = 2
      RETURN
   20 MT = MM
      REWIND MT
      NIN = NI
      REWIND NIN
      NOUT = NO
      REWIND NOUT
      MP1 = M + 1
      NN = N
      NEL = NPM
C
C - - CALCULATE THE MAXIMUM NO. OF ROWS, 'K'
C
   30 K = (KORE - NEL) / NEL
C
C - - TEST TO SEE IF THE REST OF THE MATRIX WILL FIT IN CORE
C
      LAST = K .GE. NN
      IF (LAST) K = NN
C
C - - READ 'K' ROWS OF THE AUGMENTED 'A' MATRIX
C
   40 NT = 0
      DO 50 IR = 1, K
      NS = NT + 1
      NT = NT + NEL
   50 CALL  GETT(NIN, 1, NEL, A(NS), 1, AA2)
C
C - - CHECK TO SEE IF WE WERE UNLUCKY ENOUGH TO END UP WITH ONLY ONE ROW
C
      IF (K .EQ. 1) GO TO 110
C
C - - 'K' IS GREATER THAN '1' SO WE CAN START THE TRIANGULARIZATION
C
      NELP1 = NEL + 1
      NS = - NEL
      NELP2 = NELP1 + 1
C
C - - FORM THE 'TRAPEZOIDAL' ARRAY  (8)
C
      DO 60 IR = 2, K
      NP = NELP2 - IR
      NS = NS + NELP1
      NT = NS
      DO 60 IO = IR, K
      NT = NT + NEL
      MN = NT
      NB = NS
      A(NT) = (-A(NT)) / A(NS)
      DO 60 NF = 2, NP
      MN = MN + 1
      NB = NB + 1
   60 A(MN) = A(MN) + A(NT) * A(NB)
      IF (LAST) GO TO 110
C
C - - WRITE THE 'TRAPEZOIDAL' MATRIX ON TAPE
C
      NT = 0
      NP = NEL
      NS = - NEL
      DO 70 IO = 1, K
      NS = NS + NELP1
      NT = NT + NEL
      CALL  SAVE(MT, 2, NP, NP, A(NS), 1, AA2)
   70 NP = NP - 1
      NP = NP - M
      NS = KORE - NEL + 1
C
C - - READ ANOTHER ROW
C
      DO 100 IO = 1, NP
      CALL  GETT(NIN, 1, NEL, A(NS), 1, AA2)
C
C - - MODIFY THIS ROW BY THE 'TRAPEZOIDAL' ARRAY
C
      NT = 1
      MN = NS
      DO 90 IR = 1, K
      NB = NT
      NF = MN + 1
      A(MN) = (-A(MN)) / A(NT)
      DO 80 NN = NF, KORE
      NB = NB + 1
   80 A(NN) = A(NN) + A(MN) * A(NB)
      MN = NF
   90 NT = NT + NELP1
C
C - - WRITE THE MODIFIED ROW ON TAPE
C
      NN1 = KORE - MN + 1
  100 CALL  SAVE(NOUT, 1, NN1, NN1, A(MN), 1, AA2)
      REWIND NOUT
      REWIND NIN
C
C - - SWITCH THE TAPES
C
      NT = NIN
      NIN = NOUT
      NOUT = NT
C
C - - RE-CALCULATE ROW LENGTH AND LOOP BACK
C
      NEL = NEL - K
      NN = NEL - M
      GO TO 30
C
C - - REWIND ALL TAPES
C
  110 REWIND MT
      REWIND NIN
      REWIND NOUT
C
C - - CONDENSE THE MATRIX
C
      NN = NEL
      NL = NEL + 1
      IF (K .EQ. 1) GO TO 130
      NS = 1
      NT = NEL
      DO 120 IR = 2, K
      NS = NS + NELP1
      NT = NT + NEL
      DO 120 IO = NS, NT
      A(NL) = A(IO)
  120 NL = NL + 1
  130 N1 = KORE - K * M + 1
C
C - - THERE, NOW WE CAN START THE BACK-SOLUTION
C * * NOTE..THE FIRST AVAILABLE LOCATION FOR THE SOLUTIONS IS A(N1)
C
      NREM = N
      NEL = NPM
      LAST = K .EQ. N
      NPASS = 0
C
C - - SOLVE FOR THE ANSWERS CORRESPONDING TO 'K' ROWS
C
  140 KM1 = K - 1
      KP1 = K + 1
      NS = NL - MP1
      NPASS = NPASS + 1
      DO 170 MN = 1, M
      NF = NS + MN
      A(NF) = A(NF) / A(NS)
      NT = NS
      IF (KM1 .EQ. 0) GO  TO 170
      DO 160 IR = 1, KM1
      NF = NF - IR - M
      NT = NT - MP1 - IR
      SUM = 0.0
      NP = NF
      N2 = MP1 + IR
      DO 150 IO = 1, IR
      NN = NT + IO
      NP = NP + N2 - IO
  150 SUM = SUM + A(NN) * A(NP)
  160 A(NF) = (A(NF) - SUM) / A(NT)
  170 CONTINUE
C
C - - MOVE THE SOLUTIONS TO CONTIGUOUS LOCATIONS STARTING AT A(N1)
C
      N1 = KORE + 1
      DO 190 NN = 1, K
      DO 180 MN = 1, M
      NL = NL - 1
      N1 = N1 - 1
  180 A(N1) = A(NL)
  190 NL = NL - NN
C
C - - WRITE THE SOLUTIONS ON TAPE
C
      WRITE (NIN) K
      NS = N1 - 1
      DO 200 MN = 1, M
      NT = NS + MN
  200 WRITE ( NIN ) (A(IO), IO = NT, KORE, M)
C
C - - TEST IF THIS IS THE LAST PASS
C
      IF (LAST) GO TO 280
C
C - - WE MUST NOW MODIFY THE TRIANGULAR MATRIX TO REFLECT THE EFFECT OF
C     THE SOLUTIONS OBTAINED SO FAR   (EQ 21)
C * - NOTE..LOCATIONS A(1) TO A(N1-1) ARE NOW FREE TO USE
C
C - - CALCULATE THE NEXT VALUES OF 'NEL' AND 'NREM'
C
      NELOLD = NEL
      KOLD = K
      NEL = NEL - K
      NREM = NREM - K
C
C     CALCULATE NEW K.   B AND C (REAL) WILL ALWAYS BE INTEGERS.
C     K WILL BE CALCULATED REAL AND TRUNCATED - - GOOD.
C
      B = 1 + 2*M
      C = 2*(KOLD*(M+1) - KORE)
      K = (-B + SQRT(B**2 - 4*C))/2.0
      NROW = NREM - K + 1
      IF (K .LT. NREM) GO TO 210
      LAST = .TRUE.
      NROW = 1
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
      K = NREM
  210 NS = 1
      NT = NELOLD + 1
C
C - - READ IN THE ROWS TO BE MODIFIED
C
      DO 270 IR = 1, NREM
      NT = NT - 1
      IF (IR .LE. NROW) GO TO 220
      NS = NS + MM
      NT = NT + NN
  220 CALL  GETT(MT, 2, MN, A(NS), 1, AA2)
      NP = NT - 1
      NE = NT - M - KM1
      NN = NN - KOLD
      DO 240 MN = 1, M
      N2 = NE
      NA = NP + MN
      NB = NA
      SUM = 0.0
      DO 230 IO = 1, KOLD
      SUM = SUM + A(N2) * A(NA)
      N2 = N2 + 1
  230 NA = NA + M
      N2 = N2 + MN - 1
  240 A(N2) = A(N2) - SUM
C
C - - WRITE THE MODIFIED ROW ON TAPE OR CONDENSE THE ROW
C
      NL = NT - M + 1
      IF (IR .GE. NROW) GO TO 250
      NE = NL - KP1
      NN1 = NE - NS + 1
      NN2 = NT - NL + 1
      CALL  SAVE(NOUT, 4, NN, NN1, A(NS), NN2, A(NL))
      GO TO 270
  250 NE = NL - KOLD
      DO 260 MN = NL, NT
      A(NE) = A(MN)
  260 NE = NE + 1
  270 CONTINUE
      REWIND MT
      REWIND NOUT
C
C - - SWITCH THE TAPES
C
      NT = MT
      MT = NOUT
      NOUT = NT
C
C - - LOOP BACK THRU THE SOLUTION
C
      NL = NE
      GO TO 140
C
C - - START TO WRAP IT UP
C
  280 REWIND NIN
      N2 = N
C
C * * NOTE.. AT THIS POINT ALL LOCATIONS A(1) THRU A(KORE) ARE FREE
C
      DO 300 IB = 1, NPASS
      READ (NIN) K
      N1 = N2 - K + 1
      NS = N1
      NT = N2
C
C - - READ IN THE SOLUTIONS
C
      NM = NT - NS + 1
      DO 290 IO = 1, M
      CALL  GETT(NIN, 1,NM, A(NS), 1, AA2)
  290 NS = NS + N
  300 N2 = N1 - 1
      IR = 1
      RETURN
      END


*DECK SAVE
      SUBROUTINE  SAVE(IU, IT, N, N1, A1, N2, A2)
C
      DIMENSION  A1(N1), A2(N2)
C
      GO TO ( 10 , 20 , 30 , 40 ), IT
C
C   WRITE A1
   10 WRITE(IU) A1
      RETURN
C
C   WRITE N AND A1
   20 WRITE(IU)  N, A1
      RETURN
C
C   WRITE A1 AND A2
   30 WRITE(IU) A1, A2
      RETURN
C
C   WRITE N, A1, AND A2
   40 WRITE(IU) N, A1, A2
      RETURN
      END


*DECK GETT
      SUBROUTINE GETT(IU, IT, N1, A1, N2, A2)
C
      DIMENSION  A1(N1), A2(N2)
C
      GO TO ( 10 , 20 , 30 , 40 ), IT
C
C   READ A1
   10 READ(IU) A1
      RETURN
C
C   READ N1 AND A1
   20 READ(IU) N1, A1
      RETURN
C
C   READ A1 AND A2
   30 READ(IU) A1, A2
      RETURN
C
C   READ IDUM AND A1
   40 READ(IU) IDUM, A1
      RETURN
      END


*DECK BAKSUB
      SUBROUTINE BAKSUB(TRANS,SUMMER,NEWMAX)
C
C   THIS SUBROUTINE BACK SUBSTITUTES THE COEFFICIENT MATRIX AND THE
C   GAMMA SOLUTION TO OBTAIN THE RIGHT SIDE MATRIX FOR THE PURPOSE OF
C   CHECKING THE MATRIX SOLUTION.
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/SOLV1/GAMMA(600,10)
      DIMENSION TRANS(NEWMAX),SUMMER(NEWMAX)
C
      WRITE(6, 10 )
   10 FORMAT(1H1,47X,26HBACK SUBSTITUTION SOLUTION)
C
C   CYCLE THE RIGHT HAND SIDES
      DO 60  N = 1,NCASES
      IREAD = 0
C
C   CYCLE THE MATRIX ROWS CORRESPONDING TO ELEMENTS ON THE WING
      DO 40 I = 1,NEWMAX
C   READ THE COEFFICIENT MATRIX ROW
      IF(I .EQ. NWT+1) IREAD = NMAX
      IREAD = IREAD + 1
   20 CALL READMS(1,TRANS,NEWMAX,IREAD)
C
C   SUM UP THE TERMS FOR THIS ROW AND RIGHT SIDE
      SUMMER(I) = 0.00
      DO 30 J = 1,NEWMAX
      SUMMER(I) = SUMMER(I) + TRANS(J) * GAMMA(J,N)
   30 CONTINUE
   40 CONTINUE
C
C   PRINT THE NTH RIGHT SIDE COLUMN
      WRITE(6, 50 ) N,SUMMER
   50 FORMAT(1H0,50X,17HRIGHT SIDE COLUMN,I4,60(/1X,10E13.5))
   60 CONTINUE
      RETURN
      END
```

```
*DECK STAGE3
      OVERLAY(OVLY,3,0)
      PROGRAM STAGE3
C
C   THIS SUBROUTINE CONTROLS CALCULATION OF ALL LOADINGS FOR THE
C   FUNDAMENTAL AND COMPOSITE CASES
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)
      COMMON/FCASE3/EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10)
      COMMON/COMPOS/FACTOR(10,24),NCC
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)
      COMMON /DERIV/ UC(40),CLQ,CMQ,CMQMC
      DIMENSION CPREAD(610),CPC(600),CPA(600),CPRC(600),CPRA(600),
     1          CPP(600)
      EQUIVALENCE (BETA(1,1),CPC(1)),(BETA(1,2),CPA(1)),
     1     (BETA(1,3),CPRC(1)),(BETA(1,4),CPRA(1)),(BETA(1,5),CPP(1))
C
      IF(LOGIC .EQ. 3) GO TO 80
C   CALCULATE AND PRINT THE LOADING FOR ALL FUNDAMENTAL CASES
   10 CALL STG3FC(NEWMAX)
      DO 30  N = 1,NCASES
      LCASE = N
   20 CALL STG3FS(CLQ,CMQ,CMQMC,DUM4,NEWMAX,NOALFA,LCASE)
   30 CONTINUE
      IF(LOGIC .EQ. 2) WRITE(6, 40  ) CLQ,CMQ,CMQMC
   40 FORMAT(1H0/ 26X,43HLIFT COEFFICIENT DERIVATIVE DUE TO PITCHING,
     1        17H ABOUT XCG, CLQ =,F10.6 /
     2        14X,51HPITCHING MOMENT COEFFICIENT DERIVATIVE ABOUT ORIGIN,
     3        33H DUE TO PITCHING ABOUT XCG, CMQ =,F10.6 /
     4        16X,42HPITCHING MOMENT COEFF DERIVATIVE ABOUT XMC,
     5        35H DUE TO PITCHING ABOUT XCG, CMQMC =, F10.6)
      CALL STG3FT
C
C   CALCULATE AND PRINT THE LOADING FOR ALL COMPOSITE CASES
      IF(NCC .LT. 1) GO TO 100
      DO 60  M = 1,NCC
      MCASE = M
   50 CALL STG3C(NEWMAX,MCASE,NOALFA)
   60 CONTINUE
   70 GO TO 100
C
C   CALCULATE AND PRINT THE COEFFICIENTS AND DERIVATIVES FOR ALL
C   FUNDAMENTAL CASES
   80 CALL FUNDER(EPS,CPC,CPA,CPRC,CPRA,CPP,DEL,CHORD,Y,DELTA,CMU,AREA,
     1        CLQ,CMQ,CMQMC,CLLP,CNP2,NW,IJ,NMAX,NJT,NEWMAX,NCASES,NOALFA,
     2        NROWS,ISYMM,XLEAD,TANLE,XMC)
C
C   CALCULATE AND PRINT THE STABILITY DERIVATIVES FOR ALL COMPOSITE CASES
   90 CALL COMDER(EPS,CPC,CPA,CPRC,CPRA,CPP,CPREAD,DEL,CHORD,Y,CMU,
     1        DELTA,AREA,CLQ,CMQ,CMQMC,CLLP,CNP2,NW,IJ,NMAX,NJT,NEWMAX,
     2        NCASES,NROWS,ISYMM,XLEAD,TANLE,XMC)
  100 CONTINUE
      END
```

```
*DECK STG3FC
      SUBROUTINE STG3FC(NEWMAX)
C
C   THIS SUBROUTINE CONTROLS CALCULATION OF CHORDWISE LOADING
C   FOR FUNDAMENTAL CASES
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/FCASE3/EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10)
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)
      COMMON/SOLV1/CP(600,10)
      DIMENSION XBB(5),CPEXP(5,10)
C
C   CALCULATE AND PRINT THE CHORDWISE LOADING OF THE FUNDAMENTAL CASES
C
C   INITIALIZE THE UNUSED VALUES OF CP FOR PRINTING
      NC1 = NCASES + 1
      IF(NC1 .GT. 10) GO TO 30
      DO 20 N = NC1,10
      DO 10 I = 1,NEWMAX
      CP(I,N) = 0.00
   10 CONTINUE
   20 CONTINUE
   30 I = 0
      II = NWT
      DO 320 K = 1,NROWS
      IF(IPRINT .GT. 0) GO TO 70
      WRITE(6, 40 )
   40 FORMAT(1H1,35X,12(4H****),1H*/
     1        36X,49H*   CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES   */
     2        36X,12(4H****),1H*)
      WRITE(6, 50 ) K,Y(K),CHORD(K),(N,N=1,10)
   50 FORMAT(1H0,35X,7HSECTION,I3,5X,3HY =,F10.6,5X,7HCHORD =,F10.6/
     1        8X,1HI,5X,2HXB,3X,9(5X,4HCASE,I2),5X,4HCASE,I3)
C
C   ON THE WING
      WRITE(6, 60 )
   60 FORMAT(1H+,4HWING)
   70 NWK = NW(K)
      DO 100 L = 1,NWK
      I = I + 1
      DO 80 N = 1,NCASES
      CP(I,N) = 2.00 * CP(I,N)
   80 CONTINUE
      IF(IPRINT .LT. 1) WRITE(6, 90 ) I,XB(I),(CP(I,N),N=1,10)
   90 FORMAT(1H ,I8,11F11.6)
  100 CONTINUE
      ILINES = NWK + 4
C
C   ON THE JET
      NJK = NJ(K)
      IF(CMU(K) .LT. 0.0001) GO TO 140
      IF(IPRINT .LT. 1) WRITE(6, 110 )
  110 FORMAT(1H ,1X,3HJET)
      DO 130 L = 1,NJK
      II = II + 1
      DO 120 N = 1,NCASES
      CP(II,N) = 2.00 * CP(II,N)
  120 CONTINUE
      IF(IPRINT .LT. 1) WRITE(6, 90 ) II,XB(II),(CP(II,N),N=1,10)
  130 CONTINUE
      ILINES = ILINES + NJK + 1
C
C   PRINT THE DETAILED LOADING ON THE SINGULAR ELEMENTS
C
C   LEADING EDGE
  140 IF(IPRINT .GT. 0) GO TO 320
      WRITE(6, 150 )
  150 FORMAT(1H0,45X,29HDETAILED LEADING EDGE LOADING)
      IP = IW(K)
      DO 160 N = 1,NCASES
      LCASE = N
      CALL EXPLE(LCASE,CP(IP,LCASE),CP(IP+1,LCASE),DEL(IP),XBB,CPEXP)
  160 CONTINUE
      DO 170 M = 1,5
      WRITE(6, 90 ) M,XBB(M),(CPEXP(M,N),N=1,NCASES)
  170 CONTINUE
      ILINES = ILINES + 7
C
C   HINGES
      IF(IHINGE .EQ. 0) GO TO 320
      J = IW(K) - 1
      DO 250 L = 1,NWK
      J = J + 1
      IF(ITYPE(J) .LT. 40) GO TO 250
      DO 180 N = 1,NCASES
      LCASE = N
      CALL EXPH1(LCASE,CP(J,N),CP(J-1,N),DEL(J-1),BETA(J,N),
     1           CHORD(K),XB(J),XBB,CPEXP)
  180 CONTINUE
      IF(ILINES .LT. 46) GO TO 200
      WRITE(6, 190 )
  190 FORMAT(1H1)
      ILINES = 1
  200 WRITE(6, 210 ) J
  210 FORMAT(1H0,42X,33HDETAILED HINGE LOADING ON ELEMENT,I4)
      DO 220 M = 1,5
      WRITE(6, 90 ) M,XBB(M),(CPEXP(M,N),N=1,NCASES)
  220 CONTINUE
      DO 230 N = 1,NCASES
      LCASE = N
      CALL EXPH2(LCASE,CP(J,N),CP(J+1,N),DEL(J),BETA(J,N),
     1           CHORD(K),XB(J),XBB,CPEXP)
  230 CONTINUE
      DO 240 M = 6,10
      WRITE(6, 90 ) M,XBB(M-5),(CPEXP(M-5,N),N=1,NCASES)
  240 CONTINUE
      ILINES = ILINES + 12
  250 CONTINUE
      IF((NJ(K) .EQ. 0) .OR.(CMU(K) .LT. 0.0001)) GO TO 320
      J = IJ(K)
      IF(ITYPE(J) .NE. 43) GO TO 320
      DO 260 N = 1,NCASES
      LCASE = N
      I1 = IW(K) + NW(K) - 1
      CALL EXPH1(LCASE,CP(J,N),CP(I1,N),DEL(I1),BETA(J,N),
     1           CHORD(K),XB(J),XBB,CPEXP)
  260 CONTINUE
      IF(ILINES .LT. 46) GO TO 270
      WRITE(6, 190 )
      ILINES = 1
  270 WRITE(6, 280 ) J
  280 FORMAT(1H0,40X,37HDETAILED JET HINGE LOADING ON ELEMENT,I4)
      DO 290 M = 1,5
      WRITE(6, 90 ) M,XBB(M),(CPEXP(M,N),N=1,NCASES)
  290 CONTINUE
      DO 300 N = 1,NCASES
      LCASE = N
      CALL EXPH2(LCASE,CP(J,N),CP(J+1,N),DEL(J),BETA(J,N),
     1           CHORD(K),XB(J),XBB,CPEXP)
  300 CONTINUE
      DO 310 M = 6,10
      WRITE(6, 90 ) M,XBB(M-5),(CPEXP(M-5,N),N=1,NCASES)
  310 CONTINUE
      ILINES = ILINES + 12
  320 CONTINUE
      RETURN
      END
```

```
*DECK STG3FS
      SUBROUTINE STG3FS(CLQ,CMQ,CMQMC,CLLP,NEWMAX,NOALFA,LC)
C
C   THIS SUBROUTINE CONTROLS CALCULATION OF ALL SPANWISE AND TOTAL
C   LOADING FOR FUNDAMENTAL CASES
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(400),ITYPE(600)
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)
      COMMON/FCASE2/TST(40,10),HL(40,10),DJ(40),ACTE(40),AC(20,40),
     1             XHP(4,40),PET(4,40),IFS(4,40),ICT(40),IHT(40),NCT,NHT
      COMMON/FCASE3/EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10)
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)
      COMMON/SOLV1/CP(600,10)
      COMMON/LOAD1/ TWIST(40),HC(40),TH(40),THETS(40),
     1             BTA(600),EP(600),CPD(600)
      COMMON/LOAD2/ CLG(40),CLMU(40),CL(40),COMU(40),CDG(40),CDI(40),
     1             CS(40),CMG(40),CMMU(40),CMT(40),CM(40),XBCP(40),XBCL(40)
      COMMON/LOAD3/ CCLG(10),CCLJ(10),CCL(10),CCMG(10),CCMJ(10),
     1     CCMT(10),CCM(10),CMGMC(10),CMJMC(10),CMTMC(10),CMMC(10),
     2     CXCP(10),CXCL(10),CCJ(10),CCDG(10),CCDJ(10),CCS(10),CCDI(10),
     3     CDITZ(10),CLLG(10),CLLJ(10),CLL(10),CNJ(10),CNI(10),CCY(10),
     4     CXCPB(10),CXCLB(10)
      COMMON/LOAD4/ CLGC(40),CLMUC(40),CLQ(40),COMUQ(40),CDGQ(40),
     1     CDIQ(40),CSQ(40),CTQ(40),CMGQ(40),CMMUQ(40),CMTQ(40),CMQ(40),
     2             XBCPQ(40),XBCLQ(40),FACT(10)
      COMMON/LOAD5/ CGAM(40),CGAMC(40),ALFINF(40),ALFINQ(40),DUMB(40)
      COMMON/LOAD7/CBGR(10),CBGL(10),CBJR(10),CBJL(10),
     1             CBR(10),CBL(10),CPBMR(10),CPBML(10),CL2R(10),CL2L(10)
C
   10 N = LC
      LCASE = LC
C
C   LOAD THOSE GEOMETRIC PARAMETERS WHICH ARE DIFFERENT FOR EACH CASE
C   INTO THEIR RESPECTIVE DUMMY ARRAYS
      DO 30  K = 1,NROWS
      TWIST(K) = TST(K,LCASE)
   20 TH(K) = THETA(K,LCASE)
      THETS(K) = THS(K,LCASE)
      HC(K) = HL(K,LCASE)
      DUMB(K) = 0.00
   30 CONTINUE
      DO 50  I = 1,NEWMAX
   40 BTA(I) = BETA(I,LCASE)
      EP(I) = EPS(I,LCASE)
      CPD(I) = CP(I,LCASE)
   50 CONTINUE
C
C   COMPUTE SECTIONAL COEFFICIENTS
      ALPHA = 0.00
      IF((LCASE .EQ. 1) .AND. (NOALFA .NE. 0)) ALPHA = 1.00
   60 CALL SLOAD(ALPHA,IJ,NW,NJ,CHORD,CMU,TH,THETS,TWIST,
     1        XB,DEL,BTA,EP,CPD,CL,CLG,CLMU,CM,CMG,CMMU,CMT,XBCP,XBCL,
     2        CDI,COMU,CDG,CS,CTC,NROWS,IHINGE)
C
C   COMPUTE SECTIONAL VORTICITY OF WING-JET SYSTEM
      CALL SLOADG(CPD,DEL,BTA,CHORD,D,CMU,NJ,IJ,CLG,CGAM,NROWS,IHINGE)
C
C   COMPUTE SECTIONAL DOWNWASH AT INFINITY
   70 CALL TREFTZ(Y,DELTA,CMU,CGAM,ALFINF,NROWS,ISYMM)
C
C   COMPUTE TOTAL LINEAR COEFFICIENTS
   80 CALL TLOAD(ALPHA,AREA,CREF,XMC,Y,DELTA,CHORD,HC,XB,XLEAD,BTA,
     1        CLG,CLMU,CMG,CMMU,CMT,CMU,CCLG(N),CCLJ(N),CCL(N),CCMG(N),
     2        CCMJ(N),CCMT(N),CCM(N),CMGMC(N),CMJMC(N),CMTMC(N),CMMC(N),
     3        CXCP(N),CXCL(N),CXCPB(N),CXCLB(N),CCJ(N),CLLG(N),CLLJ(N),
     4        CLL(N),ITYPE,IW,NW,NROWS,ISYMM,
     5        CBGR(N),CBGL(N),CBJR(N),CBJL(N),CBR(N),CBL(N),
     6        CPBMR(N),CPBML(N),CL2R(N),CL2L(N))
C
C   COMPUTE TOTAL NONLINEAR COEFFICIENTS
   90 CALL TLOADX(AREA,CHORD,DELTA,Y,CMU,CDG,COMU,CS,CDI,CL,DUMB,
     1 CCDG(N),CCDJ(N),CCS(N),CCDI(N),CDITZ(N),DUMB,ALFINF,DUMMY,CCJ(1),
     2        CNJ(N),CNI(N),CCY(N),XLEAD,TANLE,XMC,NROWS,ISYMM)
C
C   DEFINE THE CONSTANT STABILITY DERIVATIVES
      CLQ = CCLG(NCASES)
      CMQ = CCMG(NCASES)
      CMQMC = CMGMC(NCASES)
      CLLP = CLLG(NCASES)
C
C   PRINT LIFT AND DRAG COEFFICIENTS
      IF(IPRINT .GT. 1) RETURN
      WRITE(6, 100 ) LCASE
  100 FORMAT(1H1,36X,11(4H****),2H**/37X,23H*  SPANWISE LOADING FOR,
     1        17H FUNDAMENTAL CASE,I3,3H  */37X,11(4H****),2H**)
      WRITE(6, 110 )
  110 FORMAT(1H ,19X,29H........... LIFT .............,5X,
     1        2H..,7(4H....),14H INDUCED DRAG , 8(4H....)/
     2        1X, 7HSECTION, 5X,1HY,8X,3HCLG,7X,4HCLMU,6X,2HCL,5X,
     3        4H  *  3X,3HCDG,8X,4HCOMU,7X,2HCS,9X,2HCD,9X,3HCMU,
     4        8X,5HGAMMA,6X,5HALFIN)
      WRITE(6, 120 ) (K,Y(K),CLG(K),CLMU(K),CL(K),CDG(K),COMU(K),
     1                CS(K),CDI(K),CMU(K),CGAM(K),ALFINF(K),K=1,NROWS)
  120 FORMAT(1H ,I4, 4X, 4F10.6, 4H  *  ,7F11.7)
      WRITE(6, 130 ) CCLG(N),CCLJ(N),CCL(N),CCDG(N),CCDJ(N),CCS(N),
     1        CCDI(N),CCJ(N),CDITZ(N)
  130 FORMAT(1H , 8X, 4(10H ---------),4H  *  ,7(11H ----------)/
     1        12X,5HTOTAL,2X,3F10.6,4H  *  ,5F11.7,11X,F11.7)
C
C   PRINT PITCHING MOMENT AND CENTER OF LIFT DATA
      IF(NROWS .GT. 21) WRITE(6, 140 )
  140 FORMAT(1H1)
      WRITE(6, 150 )
  150 FORMAT(1H0,24X,39H........... PITCHING MOMENT ............10X,
     1        20H ... LIFT CENTER .../
     2        6X, 7HSECTION,5X,1HY,8X,3HCMG,7X,4HCMMU,6X,3HCMT,7X,2HCM,
     3        5X,10H  *   *   ,3X,5HXCP/C,5X,5HXCL/C)
      WRITE(6, 160 ) (K,Y(K),CMG(K),CMMU(K),CMT(K),CM(K),XBCP(K),XBCL(K)
     1,       K=1,NROWS)
  160 FORMAT(1H ,I9,4X,5F10.6,10H   *   *   ,2F10.6)
      WRITE(6, 170 ) CCMG(N),CCMJ(N),CCMT(N),CCM(N),CXCP(N),CXCL(N),
     1        CMGMC(N),CMJMC(N),CMTMC(N),CMMC(N),CXCPB(N),CXCLB(N)
  170 FORMAT(1H ,13X, 5(10H ---------),10H   *   *   , 2(10H ---------)/
     1       17X,5HTOTAL,2X, 4F10.6, 10H  (APEX)   ,2F10.6,9H (X/CREF)/
     2       24X,4F10.6,10H  (XMC)    ,2F10.6,8H (X/B/2))
      RETURN
      END
```

```
*DECK STG3FT
      SUBROUTINE STG3FT
C
C   THIS SUBROUTINE PRINTS A TABLE OF ALL TOTAL COEFFICIENTS
C   FOR ALL FUNDAMENTAL CASES
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/LOAD3/ CCLG(10),CCLJ(10),CCL(10),CCMG(10),CCMJ(10),
     1     CCMT(10),CCM(10),CMGMC(10),CMJMC(10),CMTMC(10),CMMC(10),
     2     CXCP(10),CXCL(10),CCJ(10),CCDG(10),CCDJ(10),CCS(10),CCDI(10),
     3     CDITZ(10),CLLG(10),CLLJ(10),CLL(10),CNJ(10),CNI(10),CCY(10),
     4     CXCPB(10),CXCLB(10)
      COMMON/LOAD6/COEFF(35)
      COMMON/LOAD7/CBGR(10),CBGL(10),CBJR(10),CBJL(10),
     1             CBR(10),CBL(10),CPBMR(10),CPBML(10),CL2R(10),CL2L(10)
C
C   INITIALIZE ALL UNUSED TOTAL COEFFICIENTS TO ZERO FOR PRINTING
      IF(NCASES .EQ. 10) GO TO 20
      NC1 = NCASES + 1
      DO 10  N = NC1,10
       CCLG(N) = 0.00
       CCLJ(N) = 0.00
        CCL(N) = 0.00
       CCDG(N) = 0.00
       CCDJ(N) = 0.00
        CCS(N) = 0.00
       CCDI(N) = 0.00
      CDITZ(N) = 0.00
        CCJ(N) = 0.00
       CCMG(N) = 0.00
       CCMJ(N) = 0.00
       CCMT(N) = 0.00
        CCM(N) = 0.00
       CXCP(N) = 0.00
       CXCL(N) = 0.00
      CXCPB(N) = 0.00
      CXCLB(N) = 0.00
      CMGMC(N) = 0.00
      CMJMC(N) = 0.00
      CMTMC(N) = 0.00
       CMMC(N) = 0.00
       CLLG(N) = 0.00
       CLLJ(N) = 0.00
        CLL(N) = 0.00
        CNJ(N) = 0.00
        CNI(N) = 0.00
        CCY(N) = 0.00
       CBGR(N) = 0.00
       CBGL(N) = 0.00
       CBJR(N) = 0.00
       CBJL(N) = 0.00
        CBR(N) = 0.00
        CBL(N) = 0.00
      CPBMR(N) = 0.00
      CPBML(N) = 0.00
   10 CONTINUE
   20 WRITE(6, 30 ) (N,N=1,10)
   30 FORMAT(1H1,41X, 9(4H****)/
     1     42X,36H*  TOTAL AERODYNAMIC COEFFICIENTS  */42X,9(4H****)//
     2     14X,9(4HCASE,I2,5X), 4HCASE,I3)
   40 FORMAT(1H0, 2X,A8,10F11.7,34(/3X,A8,10F11.7))
      WRITE(6, 40 )COEFF( 1), CCLG, COEFF( 2), CCLJ, COEFF( 3),  CCL,
     1             COEFF( 4), CCDG, COEFF( 5), CCDJ, COEFF( 6),  CCS,
     2             COEFF( 7), CCDI, COEFF( 8),CDITZ, COEFF( 9),  CCJ,
     3             COEFF(10), CCMG, COEFF(11), CCMJ, COEFF(12), CCMT,
     4             COEFF(13),  CCM, COEFF(14), CXCP, COEFF(15), CXCL,
     5             COEFF(16),CXCPB, COEFF(17),CXCLB, COEFF(18),CMGMC,
     6             COEFF(19),CMJMC, COEFF(20),CMTMC, COEFF(21), CMMC,
     7             COEFF(22), CLLG, COEFF(23), CLLJ, COEFF(24),  CLL,
     8             COEFF(25),  CNJ, COEFF(26),  CNI,  COEFF(27), CCY,
     9        COEFF(28),CBGR,COEFF(29),CBGL,COEFF(30),CBJR,
     *        COEFF(31),CBJL,COEFF(32), CBR,COEFF(33),CBL,
     1        COEFF(34),CPBMR,COEFF(35),CPBML
C
   50 RETURN
      END
```

```
*DECK STG3C
      SUBROUTINE STG3C(NEWMAX,M,NOALFA)
C
C  THIS SUBROUTINE CONTROLS CALCULATION OF CHORDWISE, SPANWISE AND
C  TOTAL LOADING FOR THE REQUIRED COMPOSITE CASES
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1          D(40),KK(600),ITYPE(600)
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)
      COMMON/FCASE2/TST(40,10),HL(40,10),DJ(40),ACTE(40),AC(20,40),
     1          XHB(4,40),BET(4,40),IES(4,40),ICT(40),IHT(40),NCT,NHT
      COMMON/FCASE3/EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10)
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)
      COMMON/SOLV1/CP(600,10)
      COMMON/COMPOS/FACTOR(10,24),NCC
      COMMON/LOAD1/ TWIST(40),HO(40),TH(40),THETS(40),
     1          BTA(600),EP(600),CPO(600)
      COMMON/LOAD2/ CLG(40),CLMU(40),CL(40),CDMU(40),CDG(40),CDI(40),
     1          CS(40),CMG(40),CMMU(40),CMT(40),CM(40),XBCP(40),XBCL(40)
      COMMON/LOAD3/ CCLG(10),CCLJ(10),CCL(10),CCMG(10),CCMJ(10),
     1     CCMT(10),CCM(10),CMGMC(10),CMJMC(10),CMTMC(10),CMMC(10),
     2     CXCP(10),CXCL(10),CCJ(10),CCDG(10),CCDJ(10),CCS(10),CCDI(10),
     3     CDITZ(10),CLLG(10),CLLJ(10),CLL(10),CNJ(10),CNI(10),CCY(10),
     4     CXCPB(10),CXCLB(10)
      COMMON/LOAD4/ CLGO(40),CLMUO(40),CLO(40),CDMUO(40),CDGO(40),
     1     CDIO(40),CSO(40),CTO(40),CMGO(40),CMMUO(40),CMTO(40),CMO(40),
     2          XBCPO(40),XBCLO(40),FACT(10)
      COMMON/LOAD5/ CGAM(40),CGAMO(40),ALFINE(40),ALFINO(40),DUMB(40)
      COMMON/LOAD6/COEFF(35)
      COMMON/LOAD7/CBGR(10),CBGL(10),CBJR(10),CBJL(10),
     1          CBR(10),CBL(10),CPBMR(10),CPBML(10),CL2R(10),CL2L(10)
      DIMENSION COMUX(40),CDGX(40),CDIX(40),CSX(40),CPA(600)
      EQUIVALENCE (CP(1),CPA(1))
C
C  CALCULATE AND PRINT THE CHORDWISE LOADING FOR ALL COMPOSITE CASES
C
      I = 0
      I = 0
      II = NWT
      ILINES = 6
      WRITE(6, 10 ) M
   10 FORMAT(1H1,36X,11(4H****),1H*/
     1          37X,39H*   CHORDWISE LOADING FOR COMPOSITE CASE,I3,3H   */
     2          37X,11(4H****),1H*)
      WRITE(6, 20 ) (N,N=1,10),(FACTOR(N,M),N=1,10)
   20 FORMAT(1H ,48X,24HFUNDAMENTAL CASE FACTORS/10X,9(5X,2HA(,I1,1H),
     1     3X),5X,2HA(,I2,1H)/10X,10F12.6)
      WRITE(6, 30 )
   30 FORMAT(1H0,7X,47H*** NOTE ***  EACH LEADING EDGE CP VALUE IS THE,
     1          43H AVERAGE VALUE OF THE SINGULAR DISTRIBUTION)
      IF(IHINGE .NE. 0) WRITE(6, 40 )
   40 FORMAT(1H ,21X,47HIF A HINGE IS DEFLECTED THE LOADING IS SINGULAR,
     1     59H AND THE CP(A=0) VALUE IS FOR THE REGULAR EVD PORTION ONLY)
      WRITE(6, 50 )
   50 FORMAT(1H ,21X,41HDO NOT PLOT THESE LOADING POINTS DIRECTLY)
      DO 220 K = 1,NROWS
C
C  ON THE WING
      NWK = NW(K)
      DO 80   L = 1,NWK
      I = I + 1
   60 CPO(I) = 0.00
      DO 70   N = 1,NCASES
      FACT(N) = FACTOR(N,M)
      CPO(I) = CPO(I) + CP(I,N) * FACT(N)
   70 CONTINUE
   80 CONTINUE
      NC1 = NCASES + 1
      IF(NC1 .GT. 10) GO TO 100
      DO 90   N = NC1,10
      FACT(N) = 0.00
   90 CONTINUE
  100 J1 = IW(K)
      J2 = IW(K) + NWK - 1
      NEXT = 3 + (2+NOALFA) * (NWK/10+1)
      IF(CMU(K) .GT. 0.0) NEXT = NEXT + (2+NOALFA) * (NJ(K)/10+1) + 1
      ILINES = ILINES + NEXT
      IF((ILINES .LT. 56) .OR. (K .EQ. 1)) GO TO 120
      WRITE(6, 110 )
  110 FORMAT(1H1)
      ILINES = 1
  120 WRITE(6, 130 ) K,Y(K),CHORD(K)
  130 FORMAT(1H0,35X,7HSECTION,I3,5X,3HY =,F10.6,5X,7HCHORD =,F10.6/
     1          2X,4HWING)
      WRITE(6, 140 ) (XB(J),J=J1,J2)
  140 FORMAT(1H ,7X,2HXB,10F12.6,3(/10X,10F12.6))
      WRITE(6, 150 ) (CPO(J),J=J1,J2)
  150 FORMAT(1H ,2X, 7HCP(A=0),10F12.6,3(/10X,10F12.6))
      IF(NOALFA .GT. 0) WRITE(6, 160 ) (CPA(J),J=J1,J2)
  160 FORMAT(1H ,2X, 7HCP(A=1),10F12.6,3(/10X,10F12.6))
C
C  ON THE JET
      NJK = NJ(K)
      IF(CMU(K) .LT. 0.0001) GO TO 220
      DO 190 L = 1,NJK
      II = II + 1
  170 CPO(II) = 0.00
      DO 180 N = 1,NCASES
      CPO(II) = CPO(II) + CP(II,N) * FACT(N)
  180 CONTINUE
  190 CONTINUE
  200 J1 = IJ(K)
      J2 = IJ(K) + NJK - 1
      WRITE(6, 210 )
  210 FORMAT(1H ,4H JET)
      WRITE(6, 140 ) (XB(J),J=J1,J2)
      WRITE(6, 150 ) (CPO(J),J=J1,J2)
      IF(NOALFA .GT. 0) WRITE(6, 160 ) (CPA(J),J=J1,J2)
  220 CONTINUE
C
C  COMPUTE AND PRINT SPANWISE AND TOTAL LOADINGS FOR EACH COMPOSITE CASE
C
C  DEFINE THE FUNDAMENTAL CASE VARIABLES FOR ALPHA = 1
      IF(NOALFA .EQ. 0) GO TO 280
      DO 240 K = 1,NROWS
  230 TWIST(K) = TST(K,1)
      TH(K) = THETA(K,1)
      THETS(K) = THS(K,1)
  240 CONTINUE
      DO 260 I = 1,NEWMAX
  250 BTA(I) = BETA(I,1)
      EP(I) = EPS(I,1)
  260 CONTINUE
C
C  COMPUTE SECTIONAL COEFFICIENTS FOR ALPHA = 1
      ALPHA = 1.0
  270 CALL SLOAD(ALPHA,IJ,NW,NJ,CHORD,CMU,TH,THETS,TWIST,
     1       XB,DEL,BTA,EP,CPA,CL,CLG,CLMU,CM,CMG,CMMU,CMT,XBCP,XBCL,
     2       CDI,CDMU,CDG,CS,CTC,NROWS,IHINGE)
C
C  COMPUTE SECTIONAL VORTICITY FOR ALPHA = 1
      CALL SLOADG(CPA,DEL,BTA,CHORD,D,CMU,NJ,IJ,CLG,CGAM,NROWS,IHINGE)
C
C  COMPUTE SECTIONAL DOWNWASH AT INFINITY FOR ALPHA = 1
      CALL TREFTZ(Y,DELTA,CMU,CGAM,ALFINE,NROWS,ISYMM)
C
C  MODULATE AND SUM THE FUNDAMENTAL CASE VARIABLES FOR ALPHA = 0
  280 DO 310 K = 1,NROWS
      TWIST(K) = 0.00
      TH(K) = 0.00
      THETS(K) = 0.00
      HO(K) = 0.00
      DO 300 N = 1,NCASES
  290 TWIST(K) = TWIST(K) + TST(K,N) * FACT(N)
      TH(K) = TH(K) + THETA(K,N) * FACT(N)
      THETS(K) = THETS(K) + THS(K,N) * FACT(N)
      HO(K) = HO(K) + HL(K,N) * FACT(N)
  300 CONTINUE
  310 CONTINUE
      DO 350 I = 1,NEWMAX
  320 BTA(I) = 0.00
      EP(I) = 0.00
      DO 340 N = 1,NCASES
  330 BTA(I) = BTA(I) + BETA(I,N) * FACT(N)
      EP(I) = EP(I) + EPS(I,N) * FACT(N)
  340 CONTINUE
  350 CONTINUE
C
C  COMPUTE SECTIONAL COEFFICIENTS FOR ALPHA = 0
      ALPHA = 0.00
      IF(NOALFA .GT. 0) ALPHA = FACT(1)
  360 CALL SLOAD(ALPHA,IJ,NW,NJ,CHORD,CMU,TH,THETS,TWIST,
     1       XB,DEL,BTA,EP,CPO,CLO,CLGO,CLMUO,CMO,CMGO,CMMUO,CMTO,
     2       XBCPO,XBCLO,CDIO,CDMUO,CDGO,CSO,CTO,NROWS,IHINGE)
C
C  COMPUTE SECTIONAL VORTICITY FOR ALPHA = 0
  370 CALL SLOADG(CPO,DEL,BTA,CHORD,D,CMU,NJ,IJ,CLGO,CGAMO,NROWS,IHINGE)
C
C  COMPUTE SECTIONAL DOWNWASH AT INFINITY FOR ALPHA = 0
  380 CALL TREFTZ(Y,DELTA,CMU,CGAMO,ALFINO,NROWS,ISYMM)
C
C  COMPUTE SECTIONAL NONLINEAR CROSS-PRODUCT COEFFICIENTS
      IF(NOALFA .EQ. 0) GO TO 400
  390 CALL SLOADX(CPA,CPO,DEL,EP,CMU,TH,NW,NJ,IJ,
     1            CLGO,CDGX,COMUX,CSX,CDIX,NROWS)
C
C  COMPUTE TOTAL LINEAR COEFFICIENTS FOR ALPHA = 0
  400 CALL TLOADO(CREF,CCLG,CCLJ,CCMG,CCMJ,CCMT,CMGMC,CMJMC,CMTMC,
     1       CLLG,CLLJ,FACT,CCLGO,CCLJO,CCLO,CCMGO,CCMJO,CCMTO,CCMO,
     2       CMGMCO,CMJMCO,CMTMCO,CMMCO,CXCPO,CXCLO,CXCPBO,CXCLBO,
     3       CLLGO,CLLJO,CLLO,NCASES,ISYMM,
     4       CBGR,CBGL,CBJR,CBJL,CBR,CBL,CPBMR,CPBML,CL2R,CL2L,
     5       CBGRO,CBGLO,CBJRO,CBJLO,CBRO,CBLO,CPBMRO,CPBMLO,CL2RO,CL2LO)
C
C  COMPUTE TOTAL NONLINEAR COEFFICIENTS FOR ALPHA = 0
  410 CALL TLOADX(AREA,CHORD,DELTA,Y,CMU,CDGO,CDMUO,CSO,CDIO,CLO,DUMB,
     1       CCDGO,CCDJO,CCSO,CCDIO,CDITZO,DUMB,ALFINO,CCTO,CCJ(1),
     2       DUMMY,CNIO,CCYO,XLEAD,TANLE,XMC,NROWS,ISYMM)
C
C  COMPUTE TOTAL NONLINEAR CROSS-PRODUCT COEFFICIENTS
      IF(NOALFA .EQ. 0) GO TO 430
  420 CALL TLOADX(AREA,CHORD,DELTA,Y,CMU,CDGX,COMUX,CSX,CDIX,CL,CLO,
     1       CCDGX,CCDJX,CCSX,CCDIX,CDITZX,ALFINE,ALFINO,DUMMY,CCJ(1),
     2       DUMMY,CNIX,CCYX,XLEAD,TANLE,XMC,NROWS,ISYMM)
C
C  PRINT THE SECTIONAL AND TOTAL COEFFICIENTS
C
  430 WRITE(6, 440 ) M
  440 FORMAT(1H1,47X,6(4H****)/48X,18H*  COMPOSITE  CASE,I3,3H  */
     1          49X,6(4H****))
      WRITE(6, 450 ) (N,N=1,10),(FACT(N),N=1,10)
  450 FORMAT(1H ,48X,24HFUNDAMENTAL CASE FACTORS/10X,9(4X,2HA(,I1,1H),
     1     2X),3X,2HA(,I2,1H)/10X,10F10.6/)
      WRITE(6, 460 )
  460 FORMAT(1H ,20X,29H........... LIFT ............,11H   *   *   .
     1       39H............ PITCHING MOMENT ...........,10H   *   *   .
     2       20H ... LIFT CENTER .../ 2X,7HSECTION,5X,1HY,8X,4HCLGO,6X,
     3       5HCLMUO,5X,3HCLO,4X,10H   *   *   ,3X,4HCMGO,6X,5HCMMUO,5X,
     4       4HCMTO,6X,3HCMO,4X,10H   *   *   ,3X,6HXCPO/C,4X,6HXCLO/C)
      IF(NOALFA .GT. 0) WRITE(6, 470 )
  470 FORMAT(1H ,
     1       22X,4HCLGA,6X,5HCLMUA,5X,3HCLA,4X,10H   *   *   ,3X,4HCMGA,
     2       6X,5HCMMUA,6X,4HCMTA,6X,3HCMA,4X,10H   *   *   ,3X,6HXCPA/C,
```

```
    3      4X,6HXCLA/C)
      DO 500 K = 1,NROWS
      WRITE(6, 480 ) K,Y(K),CLGC(K),CLMUC(K),CLC(K),CMGC(K),CMMUC(K),
     1      CMTC(K),CMC(K),XBCPC(K),XBCLC(K)
  480 FORMAT(1H ,I5,4X,4F10.6,10H   *  *   ,4F10.6,10H   *  *   ,2F10.6)
      IF(NOALFA .GT. 0) WRITE(6, 490 ) CLG(K),CLMU(K),CL(K),
     1      CMG(K),CMMU(K),CMT(K),CM(K),XBCP(K),XBCL(K)
  490 FORMAT(1H ,19X,3F10.6,10H   *  *   ,4F10.6,10H   *  *   ,2F10.6)
  500 CONTINUE
      WRITE(6, 510 ) CCLGC,CCLJC,CCLC,CCMGC,CCMJC,CCMTC,CCMC,CXCPC,CXCLC
  510 FORMAT(1H , 9X,4(10H ---------),10H   *  *   ,4(10H ---------),
     1      10H   *  *   ,2(10H ---------)/13X,5HTOTAL,2X,3F10.6,
     2      10H   *  *   ,4F10.6,10H  (APEX)   , 2F10.6)
      IF(NOALFA .GT. 0) WRITE(6, 520 ) CCLG(1),CCLJ(1),CCL(1),
     1      CCMG(1),CCMJ(1),CCMT(1),CCM(1),CXCP(1),CXCL(1)
  520 FORMAT(1H ,19X,3F10.6,
     1      10H   *  *   ,4F10.6,10H  (APEX)   , 2F10.6)
      WRITE(6, 530 ) CMGMCC,CMJMCC,CMTMCC,CMMCC,CXCPBC,CXCLBC
  530 FORMAT(1H ,49X,
     1      10H   *  *   ,4F10.6,10H  (XMC)    , 2F10.6)
      IF(NOALFA .GT. 0) WRITE(6, 540 )
     1      CMGMC(1),CMJMC(1),CMTMC(1),CMMC(1),CXCPB(1),CXCLB(1)
  540 FORMAT(1H ,49X,10H   *  *   ,4F10.6,10H  (XMC)    , 2F10.6)
      WRITE(6, 550 )
  550 FORMAT(1H1,25X,9(4H....),15H INDUCED DRAG ,,9(4H....)/
     1      7X,7HSECTION,6X,1HY,7X,4HCDGC,7X,5HCDMUC,6X,3HCSC,8X,3HCDC,
     2      8X,6HGAMMAC,3X,5HALFINC,5X,3HCTC,8X,3HCMU)
      IF(NOALFA .GT. 0) WRITE(6, 560 )
  560 FORMAT(1H ,27X,4HCDGX,7X,5HCDMUX,6X,3HCSX,8X,3HCDX/
     1      28X,5HCDGA2,6X,6HCDMUA2,5X,4HCSA2,7X,4HCDA2,7X,6HGAMMA2,
     2      5X,6HALFINA)
      DO 590 K = 1,NROWS
      WRITE(6, 570 ) K,Y(K),CDGC(K),CDMUC(K),CSC(K),CDIC(K),
     1      CGAMC(K),ALFINC(K),CTC(K),CMU(K)
  570 FORMAT(1H , 6X,I4 ,4X,F10.6,8F11.7)
      IF(NOALFA .GT. 0) WRITE(6, 590 )
     1      CDGX(K),CDMUX(K),CSX(K),CDIX(K),CDG(K),CDMU(K),CS(K),
     2      CDI(K),CGAM(K),ALFINF(K)
  580 FORMAT(1H ,24X,4F11.7/25X,6F11.7)
  590 CONTINUE
      WRITE(6, 600 ) CCDGO,CCDJO,CCSO,CCDIO,CDITZO,CCTO,CCJ(1)
  600 FORMAT(1H ,24X,8(11H ----------)/18X,5HTOTAL,2X,4F11.7,11X,3F11.7)
      IF(NOALFA .GT. 0) WRITE(6, 610 ) CCDGX,
     1   CCDJX,CCSX,CCDIX,CDITZX,CCDG(1),CCDJ(1),CCS(1),CCDI(1),CDITZ(1)
  610 FORMAT(1H ,24X,4F11.7,11X,F11.7/25X,4F11.7,11X,F11.7)
C
C  PRINT A TABLE OF ALL TOTAL COEFFICIENTS FOR ALPHA = 0,ALPHA,ALPHA**2
C
      WRITE(6, 620 )
  620 FORMAT(1H1,41X,9(4H****)/42X,20H*   TOTAL AERODYNAMIC,
     1      16H COEFFICIENTS   */42X,9(4H****)//
     2      42X, 7HALPHA=0,   8X, 5HALPHA,10X, 8HALPHA**2)
      IF(NOALFA .GT. 0) GO TO 660
  630 FORMAT(1H ,25X,A8,F15.7)
      WRITE(6, 630 ) COEFF(1),CCLGO,COEFF(2),CCLJO,COEFF(3),CCL0,
     1      COEFF(4),CCDGO,COEFF(5),CCDJO,COEFF(6),CCSO,COEFF(7),CCDIO,
     2      COEFF(8),CDITZO,COEFF(9),CCJ(1),COEFF(10),CCMGO,
     3      COEFF(11),CCMJO,COEFF(12),CCMTO,COEFF(13),CCMO,
     4      COEFF(14),CXCPO,COEFF(15),CXCLO,COEFF(16),CXCPB,
     5      COEFF(17),CXCLB,COEFF(18),CMGMCO,COEFF(19),CMJMCO,
     6      COEFF(20),CMTMCO,COEFF(21),CMMCO,COEFF(22),CLLGO,
     7      COEFF(23),CLLJO,COEFF(24),CLLO,COEFF(25),CNJ(1),
     8      COEFF(26),CNIO,COEFF(27),CCYO,
     9      COEFF(28),CBGRO,COEFF(29),CBGLO,COEFF(30),CBJRO,
     *      COEFF(31),CBJLO,COEFF(32),CBRO, COEFF(33),CBLO,
     1      COEFF(34),CPBMRO,COEFF(35),CPBMLO
      RETURN
C
  640 FORMAT(1H ,25X,A8,3F15.6)
  650 FORMAT(1H ,25X,A8,1X,3F15.7)
  660 WRITE(6, 640 )COEFF( 1), CCLGO,CCLG(1)
      WRITE(6, 640 )COEFF( 2), CCLJO,CCLJ(1)
      WRITE(6, 640 )COEFF( 3), CCLO ,CCL(1)
      WRITE(6, 650 )COEFF( 4), CCDGO,CCDGX,CCDG(1)
      WRITE(6, 650 )COEFF( 5), CCDJO,CCDJX,CCDJ(1)
      WRITE(6, 650 )COEFF( 6), CCSO ,CCSX ,CCS(1)
      WRITE(6, 650 )COEFF( 7), CCDIO,CCDIX,CCDI(1)
      WRITE(6, 650 )COEFF( 8), CDITZO,CDITZX,CDITZ(1)
      WRITE(6, 650 )COEFF( 9), CCJ(1)
      WRITE(6, 640 )COEFF(10), CCMGO,CCMG(1)
      WRITE(6, 640 )COEFF(11), CCMJO,CCMJ(1)
      WRITE(6, 640 )COEFF(12), CCMTO,CCMT(1)
      WRITE(6, 640 )COEFF(13), CCMO ,CCM(1)
      WRITE(6, 640 )COEFF(14), CXCPO,CXCP(1)
      WRITE(6, 640 )COEFF(15), CXCLO,CXCL(1)
      WRITE(6, 640 )COEFF(16), CXCPBO,CXCPB(1)
      WRITE(6, 640 )COEFF(17), CXCLBO,CXCLB(1)
      WRITE(6, 640 )COEFF(18), CMGMCO,CMGMC(1)
      WRITE(6, 640 )COEFF(19), CMJMCO,CMJMC(1)
      WRITE(6, 640 )COEFF(20), CMTMCO,CMTMC(1)
      WRITE(6, 640 )COEFF(21), CMMCO,CMMC(1)
      WRITE(6, 640 )COEFF(22), CLLGO,CLLG(1)
      WRITE(6, 640 )COEFF(23), CLLJO,CLLJ(1)
      WRITE(6, 640 )COEFF(24), CLLO,CLL(1)
      WRITE(6, 640 )COEFF(25), CNJ(1)
      WRITE(6, 640 )COEFF(26), CNIO,CNIX,CNI(1)
      WRITE(6, 640 )COEFF(27), CCYO,CCYX,CCY(1)
      WRITE(6, 640 )COEFF(28), CBGRO,  CBGR(1)
      WRITE(6, 640 )COEFF(29), CBGLO,  CBGL(1)
      WRITE(6, 640 )COEFF(30), CBJRO,  CBJR(1)
      WRITE(6, 640 )COEFF(31), CBJLO,  CBJL(1)
      WRITE(6, 640 )COEFF(32), CBRO,   CBR(1)
      WRITE(6, 640 )COEFF(33), CBLO,   CBL(1)
      WRITE(6, 640 )COEFF(34), CPBMRO, CPBMR(1)
      WRITE(6, 640 )COEFF(35), CPBMLO, CPBML(1)
C
C  COMPUTE AND PRINT A TABLE OF THE VARIATION OF THE TOTAL COEFFICIENTS
C  WITH ANGLE OF ATTACK
  670 CALL TABLE(CCLO,CCL(1),CMMCC,CMMC(1),CLLO,CLL(1),CDITZO,CDITZX,
     1            CDITZ(1),CCJ(1),CNIO,CNIX,CNI(1),CNJ(1),
     2            CCYO,CCYX,CCY(1),M)
C
      RETURN
      END
```

```
*DECK DATA
      BLOCK DATA
      COMMON/LOAD6/COEFF(35)
      DATA COEFF/8H    CCLG,8H    CCLJ,8H**   CCL,8H    CCDG,8H    CCDJ,
     1           8H     CCS,8H     CCD,8H** CDITZ,8H**   CCJ,8H    CCMG,
     2           8H    CCMJ,8H    CCMT,8H     CCM,8H    CXCP,8H    CXCL,
     3           8H   CXCPB,8H   CXCLB,8H  CCMGMC,8H  CCMJMC,8H  CCMTMC,
     4           8H**  CCMMC,8H    CLLG,8H    CLLJ,8H*    CLL,8H     CNJ,
     5           8H*  CNIMC,8H*     CCY,
     6           8H    CBGR,8H    CBGL,8H    CBJR,8H    CBJL,
     7           8H     CBR,8H     CBL,8H   CPMBR,8H   CPMBL/
      END
```

```
*DECK EXPLE
      SUBROUTINE EXPLE(LCASE,CPI,CPI1,DEL,XBB,CPEXP)
C
C  THIS SUBROUTINE COMPUTES THE CP VALUE OF A LEADING EDGE EVD
C  AT 5 INTERMEDIATE POINTS ON THE ELEMENT
C
      DIMENSION XBB(5),CPEXP(5,10)
C
   10 DN = 0.20
      DO 40 N = 1,5
   20 X = DN * N
      XBB(N) = X * DEL
   30 CPEXP(N,LCASE) = 0.666666*CPI*(1.0/SQRT(X)-X) + CPI1*X
   40 CONTINUE
      RETURN
      END
```

```
*DECK EXPH1
      SUBROUTINE EXPH1(LCASE,CPI,CPI1,DEL,BTA,C,XB,XBB,CPEXP)
C
C  THIS SUBROUTINE COMPUTES THE CP VALUE OF THE FORWARD HALF OF A HINGE
C  EVD AT 5 INTERMEDIATE POINTS ON THE ELEMENT
C
      DIMENSION XBB(5),CPEXP(5,10)
C
      DN = 0.20
      DO 30  N = 1,5
   10 X = XB - DEL + (N-1)*DN*DEL
      DX = X - XB
      XBB(N) = X
   20 CPEXP(N,LCASE) = -1.273240*BTA/57.295779*(ALOG(-C*DX)
     1      + ALOG(C*DEL)*DX/DEL) +(CPI + (CPI-CPI1)*DX/DEL)
   30 CONTINUE
      RETURN
      END
```

```
*DECK EXPH2
      SUBROUTINE EXPH2(LCASE,CPI,CPI1,DEL,BTA,C,XB,XBB,CPEXP)
C
C  THIS SUBROUTINE COMPUTES THE CP VALUE OF THE REAR HALF OF A HINGE
C  EVD AT 5 INTERMEDIATE POINTS ON THE ELEMENT
C
      DIMENSION XBB(5),CPEXP(5,10)
C
      DN = 0.20
      DO 30  N = 1,5
   10 DX = N*DN*DEL
      X = XB + DX
      XBB(N) = X
   20 CPEXP(N,LCASE) = -1.273240*BTA/57.295779*(ALOG(C*DX)
     1      - ALOG(C*DEL)*DX/DEL) + (CPI - (CPI-CPI1)*DX/DEL)
   30 CONTINUE
      RETURN
      END
```

```
*DECK SLOAD
      SUBROUTINE SLOAD(ALPHA,IJ,NW,NJ,CHORD,CMU,THETA,THETAS,TST,
     1   XB,DEL,BETA,EPS,CP,CL,CLG,CLMU,CM,CMG,CMMU,CMT,XBCP,XBCL,
     2   CDI,CDMU,CDG,CS,CT,NROWS,IHINGE)
C
C  THIS SUBROUTINE COMPUTES THE SPANWISE VARIATION OF LIFT, PITCHING
C  MOMENT, AND INDUCED DRAG FOR EITHER A FUNDAMENTAL OR A COMPOSITE CASE
C
      DIMENSION IJ(40),NW(40),NJ(40)
      DIMENSION CHORD(40),CMU(40),THETA(40),THETAS(40),TST(40)
      DIMENSION CP(600),XB(600),DEL(600),BETA(600),EPS(600)
      DIMENSION CLG(40),CLMU(40),CL(40),CMG(40),CMMU(40),CMT(40),CM(40),
     1   XBCP(40),XBCL(40),CDG(40),CDMU(40),CS(40),CDI(40),CT(40)
C
C  INTEGRATE THE CHORDWISE PRESSURES FOR EACH SPANWISE SECTION
   10 I = 0
      IF(IHINGE .GT. 1) IHINGE = 1
      DO 150 K = 1,NROWS
C
C  LEADING EDGE CONTRIBUTIONS
      I = I + 1
   20 CLI = DEL(I) * (CP(I)+0.50*CP(I+1))
      CLG(K) = CLI
      CMG(K) = -DEL(I)**2 * (0.666667*CP(I)+CP(I+1)) / 3.0
      CDG(K) = CLI * EPS(I)/57.295779
   30 CS(K) = 0.1745329 * DEL(I) * CP(I)**2
C
      BCF = 0.00
      NWK = NW(K)
      DO 100 L = 2,NWK
      I = I + 1
      CPI = CP(I)
      IF(L .EQ. NWK) GO TO 40
      CPI1 = CP(I+1)
      GO TO 50
C  DEFINE TRAILING EDGE CP VALUE
   40 CPI1 = 0.0
      IF((NJ(K) .EQ. 0) .OR. (CMU(K) .LT. 0.0001)) GO TO 50
      IJK = IJ(K)
      CPI1 = CP(IJK)
C
C  REGULAR EVD CONTRIBUTIONS
   50 CLI = 0.50 * DEL(I) * (CPI+CPI1)
      CLG(K) = CLG(K) + CLI
      CMG(K) = CMG(K) - CLI*XB(I) - (CPI+2.0*CPI1)*DEL(I)**2/6.00
   60 CDG(K) = CDG(K) + CLI * EPS(I)/57.295779
      BCF = BCF + BETA(I) * (1.0-XB(I))
C
C  HINGE CONTRIBUTIONS
      IF(IHINGE .EQ. 0) GO TO 100
      B2 = BETA(I+1)
      IF(L .LT. NWK) GO TO 70
      B2 = 0.00
      IJK = IJ(K)
      IF(CMU(K) .GT. 0.0001) B2 = BETA(IJK)
   70 IF((ABS(BETA(I)) .LT. 0.0001).AND.(ABS(B2) .LT. 0.0001)) GO TO 100
      CLI = 0.00
      CMI = 0.00
      DL = ALOG(DEL(I) * CHORD(K))
      CON = 0.6366198 * DEL(I) / 57.295779
      IF(ABS(BETA(I)) .LT. 0.0001) GO TO 80
      CLI = CON * BETA(I) * (2.00 - DL)
      CMI = BETA(I) * (0.50-DL/3.00)
   80 IF(ABS(B2) .LT. 0.0001) GO TO 90
      CLI = CLI + CON * B2 * (2.00 - DL)
      CMI = CMI + 2.00 * B2 * (0.7500 - DL/3.00)
   90 CLG(K) = CLG(K) + CLI
      CMG(K) = CMG(K) - CON * DEL(I) * CMI - CLI * XB(I)
      CDG(K) = CDG(K) + CLI * EPS(I)/57.295779
  100 CONTINUE
C
C  COMPUTE THE SECTIONAL COEFFICIENTS
  110 CLMU(K) = CMU(K) * THETA(K)/57.295779
      CL(K) = CLG(K) + CLMU(K)
  120 CMMU(K) = -CMU(K) * THETA(K)/57.295779
      CMT(K) = CMU(K) * (ALPHA+TST(K)-THETAS(K)+BCF)/57.295779
      CM(K) = CMG(K) + CMMU(K) + CMT(K)
      XBCP(K) = 0.00
      XBCL(K) = 0.00
  130 IF(CLG(K) .NE. 0.00) XBCP(K) = -CMG(K) / CLG(K)
      IF(CL(K) .NE. 0.00) XBCL(K) = -(CMG(K)+CMMU(K)) / CL(K)
  140 CDMU(K) = CMU(K) * (THETA(K)/57.295779)**2 / 2.00
      CDI(K) = CDG(K) + CDMU(K) - CS(K)
      CT(K) = CMU(K) - CDI(K)
  150 CONTINUE
  160 RETURN
      END
```

```
*DECK SLOADX
      SUBROUTINE SLOADX(CPA,CPC,DEL,EPS,CMU,TH,NW,NJ,IJ,
     1          CLGO,CDGX,CDMUX,CSX,CDIX,NROWS)
C
C  THIS SUBROUTINE CALCULATES THE SECTIONAL CROSS-PRODUCT VALUES
C  OF THE NONLINEAR DRAG COEFFICIENTS
C
      DIMENSION CPA(600),CPC(600),DEL(600),EPS(600)
      DIMENSION CMU(40),TH(40),NW(40),NJ(40),IJ(40)
      DIMENSION CLGO(40),CDGX(40),CDMUX(40),CSX(40),CDIX(40)
C
      I = 0
      DO 70  K = 1,NROWS
C  LEADING EDGE EVD CONTRIBUTION
      I = I + 1
   10 CDGX(K) = DEL(I)*(CPA(I)+0.50*CPA(I+1))*EPS(I)/57.295779
   20 CSX(K) = 0.3490658 * DEL(I) * (CPC(I) * CPA(I))
      NWK = NW(K)
      DO 50 L = 2,NWK
      I = I + 1
   30 CPI1 = CPA(I+1)
C  DEFINE TRAILING EDGE CP VALUE
      IF(L .LT. NWK) GO TO 40
      CPI1 = 0.0
      IF((NJ(K) .EQ. 0) .OR. (CMU(K) .LT. 0.0001)) GO TO 40
      IJK = IJ(K)
      CPI1 = CPA(IJK)
C  REGULAR EVD CONTRIBUTION
   40 CDGX(K) = CDGX(K) + 0.50*DEL(I)*(CPA(I)+CPI1)*EPS(I)/57.295779
   50 CONTINUE
C
C  COMPUTE THE REMAINING SECTIONAL COEFFICIENTS
   60 CDGX(K) = CDGX(K) + CLGO(K)/57.295779
      CDMUX(K) = CMU(K) * TH(K)/57.295779**2
      CDIX(K) = CDGX(K) + CDMUX(K) - CSX(K)
   70 CONTINUE
      RETURN
      END
```

```
*DECK SLOADG
      SUBROUTINE SLOADG(CP,DEL,BETA,CHORD,D,CMU,NJ,IJ,CLG,CGAM,
     1          NROWS,IHINGE)
C
C  THIS SUBROUTINE COMPUTES THE SPANWISE VARIATION OF TOTAL VORTICITY
C  ON THE WING-JET SYSTEM
C
      DIMENSION CP(600),DEL(600),BETA(600)
      DIMENSION CHORD(40),D(40),CMU(40),NJ(40),IJ(40)
      DIMENSION CLG(40),CGAM(40)
C
      DO 60  K = 1,NROWS
C
C  COMPUTE THE SECTIONAL JET VORTICITY, INTEDRATED FROM T.E. TO INFINITY
   10 CGAM(K) = 0.00
      IF(CMU(K) .LT. 0.0001) GO TO 50
      II = IJ(K)
C  HINGE EVD CONTRIBUTION
      IF(IHINGE .EQ. 0) GO TO 20
      IF(BETA(II) .NE. 0.00) CGAM(K) = 0.6366198 * DEL(II) *
     1     BETA(II)/57.295779 * (2.00-ALOG(CHORD(K)*DEL(II)))
C  REGULAR EVD CONTRIBUTION
   20 NJK1 = NJ(K) - 1
      II = II - 1
      DO 40 L = 1,NJK1
      II = II + 1
   30 CGAM(K) = CGAM(K) + 0.50 * DEL(II) * (CP(II)+CP(II+1))
   40 CONTINUE
C  FAR-JET EVD CONTRIBUTION
      II = II + 1
      CGAM(K) = CGAM(K) + 0.50 * D(K) / CHORD(K) * CP(II)
C
C  SUM UP THE WING AND JET CONTRIBUTIONS
   50 CGAM(K) = 0.50 * CHORD(K) * (CGAM(K)+CLG(K))
   60 CONTINUE
      RETURN
      END
```


THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
*DECK TLOAD
      SUBROUTINE TLOAD(ALPHA,AREA,CREF,XMC,Y,DELTA,C,HC,XB,XLEAD,BETA,
     1     CLG,CLMU,CMG,CMMU,CMT,CMU,CCLG,CCLJ,CCL,CCMG,CCMJ,CCMT,CCM,
     2     CMGMC,CMJMC,CMTMC,CMMC,CXCP,CXCL,CXCPB,CXCLB,CCJ,CLLG,CLLJ,
     3     CLL,ITYPE,IW,NW,NROWS,ISYMM,
     4     CBGR,CBGL,CBJR,CBJL,CBR,CBL,CPBMR,CPBML,CL2R,CL2L)
C
C  THIS SUBROUTINE CALCULATES ALL OF THE TOTAL LOADING PARAMETERS
C  FOR A FUNDAMENTAL CASE
C
      DIMENSION Y(40),DELTA(40),C(40),HC(40),XLEAD(40),IW(40),NW(40)
      DIMENSION XB(600),BETA(600),ITYPE(600)
      DIMENSION CLG(40),CLMU(40),CMG(40),CMMU(40),CMT(40),CMU(40)
C
C  INITIALIZE THE TOTAL COEFFICIENTS
   10 CCLG = 0.00
      CCLJ = 0.00
      CCMG = 0.00
      CCMJ = 0.00
      CCMT = 0.00
      CXCP = 0.00
      CXCL = 0.00
      CXCPB = 0.00
      CXCLB = 0.00
      CCJ  = 0.00
      CLLG = 0.00
      CLLJ = 0.00
      CLL  = 0.00
      CBGR = 0.00
      CBGL = 0.00
      CBJR = 0.00
      CBJL = 0.00
      CBR = 0.00
      CBL = 0.00
      CL2R = 0.00
      CL2L = 0.00
      CPBMR = 0.00
      CPBML = 0.00
C
C  INTEGRATE THE SECTIONAL VALUES OVER THE SPAN
      DO 120 K = 1,NROWS
   20 CDEL = C(K) * DELTA(K)
      IF(ISYMM .LT. 0) GO TO 80
C
C  LIFT COEFFICIENTS
      CCLG = CCLG + CDEL * CLG(K)
   30 CCLJ = CCLJ + CDEL * CLMU(K)
C
C  PITCHING MOMENT COEFFICIENTS
      CCDEL = CDEL * C(K)
      XLB = XLEAD(K) / C(K)
C  COMPUTE LEADING EDGE HEIGHT ABOVE WING APEX
   40 I = IW(K) - 1
      NWK = NW(K)
      XDS = 0.00
      DO 60 L = 1,NWK
      I = I + 1
      IF(ITYPE(I) - 41) 60 , 50 , 70
   50 XDS = XDS + XB(I) * BETA(I)/57.295779
   60 CONTINUE
   70 HLB = HC(K) - XLB * ALPHA/57.295779 - XDS
      CCMG = CCMG + CCDEL * (CMG(K) - CLG(K)*XLB)
      CCMJ = CCMJ + CCDEL * (CMMU(K) - CLMU(K)*XLB)
      CCMT = CCMT + CCDEL * (CMT(K) - CMU(K)*HLB)
   80 CCJ  = CCJ  + CDEL * CMU(K)
C
C  ROLLING MOMENT COEFFICIENTS AND ROOT BENDING MOMENTS
      CDELY = CDEL * Y(K)
   90 IF(ISYMM .EQ. 0) GO TO 100
      CLLG = CLLG + CDELY * CLG(K)
      CLLJ = CLLJ + CDELY * CLMU(K)
  100 IF(Y(K) .LT. 0.0) GO TO 110
      CBGR = CBGR + CDELY * CLG(K)
      CBJR = CBJR + CDELY * CLMU(K)
      CL2R = CL2R + CDEL * (CLG(K)+CLMU(K))
      GO TO 120
  110 CBGL = CBGL - CDELY * CLG(K)
      CBJL = CBJL - CDELY * CLMU(K)
      CL2L = CL2L + CDEL * (CLG(K)+CLMU(K))
  120 CONTINUE
C
C  COMPUTE THE FINAL VALUES OF ALL THE TOTAL COEFFICIENTS
      FACTOR = 2.00 / AREA
      IF(ISYMM .LT. 1) FACTOR = 4.00 / AREA
  130 CCLG = FACTOR * CCLG
      CCLJ = FACTOR * CCLJ
      CCL  = CCLG + CCLJ
  140 CCJ  = FACTOR * CCJ
  150 FACTOR = FACTOR / CREF
      CCMG = FACTOR * CCMG
      CCMJ = FACTOR * CCMJ
      CCMT = FACTOR * CCMT
      CCM  = CCMG + CCMJ + CCMT
      IF(ISYMM .LT. 0) GO TO 160
      IF(CCLG .NE. 0.00) CXCP = - CCMG / CCLG
      IF(CCL .NE. 0.00) CXCL = -(CCMG+CCMJ) / CCL
      CXCPB = CXCP * CREF
      CXCLB = CXCL * CREF
  160 FACTOR = XMC / CREF
      IF(ISYMM .LT. 0) FACTOR = 0.00
      CMGMC = CCMG + CCLG * FACTOR
      CMJMC = CCMJ + CCLJ * FACTOR
      CMTMC = CCMT - CCJ  * FACTOR * ALPHA/57.295779
      CMMC  = CMGMC + CMJMC + CMTMC
  170 IF(ISYMM .EQ. 0) GO TO 180
      FACTOR = -1.00 / AREA
      IF(ISYMM .LT. 0) FACTOR = -2.00 / AREA
      CLLG = FACTOR * CLLG
      CLLJ = FACTOR * CLLJ
      CLL  = CLLG + CLLJ
  180 FACTOR = 4.0 / AREA
      CBGR = FACTOR * CBGR
      CBJR = FACTOR * CBJR
      CL2R = FACTOR * CL2R
      IF(ISYMM .GT. 0) GO TO 190
      IF(ISYMM .LT. 0) GO TO 200
      CBGL = CBGR
      CBJL = CBJR
      CL2L = CL2R
      GO TO 210
  190 CBGL = FACTOR * CBGL
      CBJL = FACTOR * CBJL
      CL2L = FACTOR * CL2L
      GO TO 210
  200 CBGL = -CBGR
      CBJL = -CBJR
      CL2L = -CL2R
  210 CBR = CBGR + CBJR
      CBL = CBGL + CBJL
      IF(CL2R .NE. 0.0) CPBMR = CBR/CL2R
      IF(CL2L .NE. 0.0) CPBML = CBL/CL2L
      RETURN
      END

*DECK TLOADX
      SUBROUTINE TLOADX(AREA,CHORD,DELTA,Y,CMU,CDG,CDMU,CS,CDI,CL,CLC,
     1                  CCDG,CCDJ,CCS,CCDI,CDITZ,ALFINE,ALFINC,CCT,CCJ,
     2                  CNJ,CNI,CCY,XLEAD,TANLE,XMC,NROWS,ISYMM)
C
C  THIS SUBROUTINE CALCULATES THE NONLINEAR TOTAL LOADING COEFFICIENTS
C  FOR ALPHA = 0 BY SPANWISE INTEGRATION OF THE NONLINEAR
C  SECTIONAL COEFFICIENTS
C
      DIMENSION CHORD(40),DELTA(40),Y(40),CMU(40),XLEAD(40),TANLE(40)
      DIMENSION CDG(40),CDMU(40),CS(40),CDI(40),CL(40),CLC(40)
      DIMENSION ALFINE(40),ALFINC(40)
C
C  INITIALIZE THE COEFFICIENTS
   10 CCDG = 0.00
      CCDJ = 0.00
      CCS = 0.00
      CDITZ = 0.00
      CNJ = 0.00
      CNI = 0.00
      CCY = 0.00
C
C  INTEGRATE THE SECTIONAL VALUES OVER THE SPAN
      DO 40  K = 1,NROWS
   20 CDEL = CHORD(K) * DELTA(K)
      CCDG = CCDG + CDEL * CDG(K)
      CCDJ = CCDJ + CDEL * CDMU(K)
   30 CCS = CCS + CDEL * CS(K)
      CDITZ = CDITZ + CDEL * (CL(K)*ALFINC(K) + CLC(K)*ALFINE(K))
      IF(ISYMM .LT. 1) GO TO 40
      CNJ = CNJ + CDEL * Y(K) * CMU(K)
      CNI = CNI + CDEL * (Y(K)*CDI(K) - CS(K)*TANLE(K)*(XLEAD(K)-XMC))
      CCY = CCY + CDEL * CS(K) * TANLE(K)
   40 CONTINUE
C  COMPUTE THE FINAL VALUES OF THE TOTAL COEFFICIENTS
      FACTOR = 2.00 / AREA
      IF(ISYMM .LT. 1) FACTOR = 4.00 / AREA
   50 CCDG = FACTOR * CCDG
      CCDJ = FACTOR * CCDJ
      CCS = FACTOR * CCS
      CDITZ = FACTOR * CDITZ / 2.00
      CCDI = CCDG + CCDJ - CCS
      CCT = CCJ - CCDI
   60 IF(ISYMM .LT. 1) RETURN
      FACTOR = 1.00 / AREA
      CNJ = -FACTOR * CNJ
      CNI = FACTOR * CNI
      CCY = FACTOR * CCY
   70 RETURN
      END
```

```
*DECK TLOAD
      SUBROUTINE TLOAD(CREF,CCLG,CCLJ,CCMG,CCMJ,CCMT,CMGMC,CMJMC,CMTMC,
     1               CLLG,CLLJ,FACT,CCLGO,CCLJO,CCLO,CCMGO,CCMJO,CCMTO,CCMO,
     2               CMGMCO,CMJMCO,CMTMCO,CMMCO,CXCPO,CXCLO,CXCPRO,CXCLRO,
     3               CLLGO,CLLJO,CLLO,NCASES,ISYMM,
     4        CBGR,CBGL,CBJR,CBJL,CBR,CBL,CPBMR,CPBML,CL2R,CL2L,
     5        CBGRO,CBGLO,CBJRO,CBJLO,CBRO,CBLO,CPBMRO,CPBMLO,CL2RO,CL2LO)
C
C  THIS SUBROUTINE CALCULATES THE LINEAR TOTAL LOADING COEFFICIENTS
C  FOR ALPHA = 0.  LINEAR QUANTITIES ARE MODULATED AND SUMMED
C  ACCORDING TO THE COMPOSITE CASE REQUIREMENTS.
C
      DIMENSION CCLG(10),CCLJ(10),CCMG(10),CCMJ(10),CCMT(10),
     1          CMGMC(10),CMJMC(10),CMTMC(10),CLLG(10),CLLJ(10),FACT(10)
      DIMENSION CBGR(10),CBGL(10),CBJR(10),CBJL(10),CBR(10),CBL(10),
     1          CPBMR(10),CPBML(10),CL2R(10),CL2L(10)
C
C  INITIALIZE THE COEFFICIENTS
   10 CCLGO = 0.00
      CCLJO = 0.00
      CCMGO = 0.00
      CCMJO = 0.00
      CCMTO = 0.00
      CMGMCO = 0.00
      CMJMCO = 0.00
      CMTMCO = 0.00
      CLLGO = 0.00
      CLLJO = 0.00
      CXCPO = 0.00
      CXCLO = 0.00
      CXCPRO = 0.00
      CXCLRO = 0.00
      CBGRO = 0.00
      CBGLO = 0.00
      CBJRO = 0.00
      CBJLO = 0.00
      CBRO = 0.00
      CBLO = 0.00
      CL2RO = 0.00
      CL2LO = 0.00
      CPBMRO = 0.00
      CPBMLO = 0.00
C
C  MODULATE AND SUM THE TOTAL COEFFICIENTS
      DO 40 N = 1,NCASES
   20 CCLGO = CCLGO + CCLG(N) * FACT(N)
      CCLJO = CCLJO + CCLJ(N) * FACT(N)
      CCMGO = CCMGO + CCMG(N) * FACT(N)
      CCMJO = CCMJO + CCMJ(N) * FACT(N)
      CCMTO = CCMTO + CCMT(N) * FACT(N)
   30 CMGMCO = CMGMCO + CMGMC(N) * FACT(N)
      CMJMCO = CMJMCO + CMJMC(N) * FACT(N)
      CMTMCO = CMTMCO + CMTMC(N) * FACT(N)
      CLLGO = CLLGO + CLLG(N) * FACT(N)
      CLLJO = CLLJO + CLLJ(N) * FACT(N)
      CBGRO = CBGRO + CBGR(N) * FACT(N)
      CBGLO = CBGLO + CBGL(N) * FACT(N)
      CBJRO = CBJRO + CBJR(N) * FACT(N)
      CBJLO = CBJLO + CBJL(N) * FACT(N)
      CL2RO = CL2RO + CL2R(N) * FACT(N)
      CL2LO = CL2LO + CL2L(N) * FACT(N)
   40 CONTINUE
C  DEFINE THE REMAINING TOTAL COEFFICIENTS
      CCLO = CCLGO + CCLJO
   50 CCMO = CCMGO + CCMJO + CCMTO
      CBRO = CBGRO + CBJRO
      CBLO = CBGLO + CBJLO
      IF(CL2RO .NE. 0.0) CPBMRO = CBRO / CL2RO
      IF(CL2LO .NE. 0.0) CPBMLO = CBLO / CL2LO
      IF(ISYMM .LT. 0) GO TO 60
      IF(CCLGO .NE. 0.00) CXCPO = - CCMGO / CCLGO
      IF(CCLO .NE. 0.00) CXCLO = - (CCMGO+CCMJO) / CCLO
      CXCPRO = CXCPO * CREF
      CXCLRO = CXCLO * CREF
   60 CMMCO = CMGMCO + CMJMCO + CMTMCO
      CLLO = CLLGO + CLLJO
   70 RETURN
      END


*DECK TREFTZ
      SUBROUTINE TREFTZ(Y,DELTA,CMU,GAMB,ALFINE,NROWS,LIKE)
      DIMENSION Y(40),DELTA(40),CMU(40),GAMB(40),ALFINE(40)
      DIMENSION E(40),B(40),C(40),DP(40),DM(40),DGAM(40)
      LOGICAL Z1,Z2,Z3,Z4,Z5,Z6,ZT1,ZT2
C
C  FIND COEFFICIENTS OF CURVES
      J=0
      DO 60 I=1,NROWS
      Z1=I.EQ.1
      Z2=LIKE.GT.0.AND.I.EQ.NROWS
      Z3=(LIKE.LE.0.AND.I.EQ.NROWS).OR.(LIKE.GT.0.AND.I.NE.NROWS.AND.
     1Y(I).GT.0.0.AND.Y(I+1).LT.0.0)
      Z4=LIKE.GT.0.AND.I.GT.1.AND.Y(I).LT.0.0.AND.Y(I-1).GT.0.0
      Z5=(I.NE.NROWS).AND.((CMU(I).LT.0.0001.AND.CMU(I+1).GT.0.0).OR.
     1(CMU(I).GT.0.0.AND.CMU(I+1).LT.0.0001))
      Z6=(I.GT.1).AND.((CMU(I).GT.0.0.AND.CMU(I-1).LT.0.0001).OR.
     1(CMU(I).LT.0.0001.AND.CMU(I-1).GT.0.0))
      D1=SQRT(1.0-ABS(Y(I)))
      ZT1=Z1.OR.Z4.OR.Z6
      ZT2=Z2.OR.Z3.OR.Z5
      IF(J.EQ.1)GO TO 10
      IF(ZT1)J=1
      IF(ZT1)GO TO 40
      J=0
   10 CONTINUE
      IF(ZT2) GO TO 30
      D2=SQRT(1.0-ABS(Y(I+1)))
      D3=SQRT(1.0-ABS(Y(I-1)))
      G1=GAMB(I)
      G2=GAMB(I+1)
      G3=GAMB(I-1)
      D13=D1**3
      D23=D2**3
      D33=D3**3
      D15=D1**5
      D25=D2**5
      D35=D3**5
      F1=D23*D35-D33*D25
      F2=D13*D35-D33*D15
      F3=D13*D25-D23*D15
      E(I)=(G1*F1-G2*F2+G3*F3)/(D1*F1-D2*F2+D3*F3)
      B(I)=((G1-E(I)*D1)*D25-(G2-E(I)*D2)*D15)/(D13*D25-D23*D15)
      C(I)=(G1-E(I)*D1-B(I)*D13)/D15
      IF(J.EQ.0) GO TO 40
   20 E(I-1)=E(I)
      B(I-1)=B(I)
      C(I-1)=C(I)
      J=0
      GO TO 40
   30 E(I)=E(I-1)
      B(I)=B(I-1)
      C(I)=C(I-1)
   40 CONTINUE
      IF(Y(I).LT.0.0) GO TO 50
      DP(I)=SQRT(ABS(D1**2-DELTA(I)))
      DM(I)=SQRT(D1**2+DELTA(I))
      GO TO 60
   50 DP(I)=SQRT(D1**2+DELTA(I))
      DM(I)=SQRT(D1**2-DELTA(I))
   60 CONTINUE
C
C  CALCULATE STRENGTH OF DISCRETE VORTICIES
      DO 80 I=1,NROWS
      IF(I.LT.NROWS.AND.((CMU(I).LT.0.0001.AND.CMU(I+1).GT.0.0).OR.
     1(CMU(I).GT.0.0.AND.CMU(I+1).LT.0.0001))) GO TO 70
      DGAM(I)=0.0
      GO TO 80
   70 DGAM(I)=(E(I+1)-E(I))*DM(I)+(B(I+1)-B(I))*DM(I)**3+(C(I+1)-C(I))
     1*DM(I)**5
   80 CONTINUE
C
C  CALCULATE DOWNWASH
      DO 100 I=1,NROWS
      ALFINE(I)=0.0
      F2=SQRT(1.0-Y(I))
      F4=SQRT(1.0+Y(I))
      DO 90 J=1,NROWS
      IF(Y(J).LT.0.0) F2=F4
      CON=1.0
      IF(Y(J).LT.0.0) CON=-1.0
      F1=ALOG(ABS(((DP(J)-F2)*(DM(J)+F2))/((DP(J)+F2)*(DM(J)-F2))))
      S1=CON*F1/F2
      S2=(2.0*(DP(J)-DM(J))+F2*F1)*CON
      S3=CON*(2.0*(DP(J)**3-DM(J)**3)/3.0)+F2**2*S2
      DELG=-DGAM(J)/(Y(I)-Y(J)+DELTA(J))
      IF(LIKE.GT.0) GO TO 90
      F3=ALOG(ABS(((DP(J)-F4)*(DM(J)+F4))/((DP(J)+F4)*(DM(J)-F4))))
      S2P=2.0*(DP(J)-DM(J))+F4*F3
      IF(LIKE.LT.0) SY=-1.0
      IF(LIKE.EQ.0) SY=1.0
      S1=S1+SY*F3/F4
      S2=S2+SY*S2P
      S3=S3+SY*((2.0*(DP(J)**3-DM(J)**3)/3.0)+F4**2*S2P)
      DELG=DELG-SY*DGAM(J)/(-Y(I)-Y(J)+DELTA(J))
   90 ALFINE(I)=ALFINE(I)+0.1591549*(0.5*E(J)*S1+1.5*B(J)*S2+2.5*C(J)*
     1S3+DELG)
  100 CONTINUE
      RETURN
      END

      SUBROUTINE TABLE(CLO,CLA,CMO,CMA,CLLO,CLLA,CDIO,CDIX,CDIA2,CJ,
     1                 CNIO,CNIX,CNIA2,CNJ,CYO,CYX,CYA2,MCASE)
C
C  THIS SUBROUTINE COMPUTES AND PRINTS A TABLE OF TOTAL COEFFICIENTS
C  FOR A RANGE OF ANGLES OF ATTACK
C
C  COMPUTE THE COEFFICIENTS
      WRITE(6, 10 ) MCASE
   10 FORMAT(1H1,31X,14(4H****)      /32X,27H*  TABULATED TOTAL COEFFICI,
     1      23HENTS FOR COMPOSITE CASE, I3, 3H  */ 32X,14(4H****)//
     2  4X,5HALPHA,10X,3HCCL,8X,6HCCL**2,5X,7HCCM(MC),4X,3HCLL,13X,
     3  5HCDITZ,6X,3HCCT,13X,3HCNI,8X,2HCN,9X,3HCCY)
      ALPHA = -11.00
      DO 60  M = 1,41
   20 ALPHA = ALPHA + 1.00
      ALPHA2 = ALPHA * ALPHA
C  LINEAR COEFFICIENTS
   30 CL = CLO + CLA * ALPHA
      CL2 = CL * CL
      CM = CMO + CMA * ALPHA
      CLL = CLLO + CLLA * ALPHA
C  NONLINEAR COEFFICIENTS
   40 CDI = CDIO + CDIX * ALPHA + CDIA2 * ALPHA2
      CT = CJ - CDI
      CNI = CNIO + CNIX * ALPHA + CNIA2 * ALPHA2
      CN = CNJ + CNI
      CY = CYO + CYX * ALPHA + CYA2 * ALPHA2
```

```
C
C  PRINT THE TABLE
      WRITE(6, 50 ) ALPHA,CL,CL2,CM,CLL,CDI,CT,CNI,CN,CY
   50 FORMAT(1H , F10.6,5H  *  ,4F11.7,5H  *  ,2F11.7,5H  *  ,3F11.7)
   60 CONTINUE
      RETURN
      END


*DECK FUNDER
      SUBROUTINE FUNDER(EPS,CPC,CPA,CPRC,CPRA,CPP,DEL,CHORD,Y,DELTA,CMU,
     1 AREA,CLQ,CMQ,CMOMC,CLLP,CNP2,NW,IJ,NMAX,NJT,NEWMAX,NCASES,NOALFA,
     2    NROWS,ISYMM,XL,TL,XMC)
C
C  THIS SUBROUTINE CONTROLS CALCULATION OF ALL AERODYNAMIC COEFFICIENTS
C  AND STABILITY DERIVATIVES FOR THE FUNDAMENTAL CASES
C
      COMMON/SOLV1/CP(600,10)
      DIMENSION CPC(NEWMAX),CPA(NEWMAX),CPRC(600),CPRA(600),CPP(600)
      DIMENSION DEL(600),EPS(600,10),EP(600)
      DIMENSION CHORD(40),Y(40),DELTA(40),CMU(40),XL(40),TL(40)
      DIMENSION NW(40),IJ(40)
C
C  FIND THE SPOT WHERE THE SOLUTION OF THE FIRST RUN WAS STORED
      IREAD = NMAX + NJT
      IFUDGE = 0
      CLLP = 0.00
      CNP2 = 0.00
      CNP0 = 0.00
C  CALCULATE THE CHORDWISE LOADING
   10 CALL STG3EC(NEWMAX)
C
C  DEFINE THE DUMMY CP ARRAYS FOR THE ALPHA CASE
C  AND THE CP ARRAY FOR THE ROLLING FUNDAMENTAL CASE
      DO 20 I = 1,NEWMAX
      CPA(I) = 0.0
      CPRA(I) = 0.0
      CPP(I) = CP(I,NCASES)
   20 CONTINUE
      DO 130 N = 1,NCASES
      LCASE = N
      IREAD = IREAD + 1
C
C  CALCULATE THE SPANWISE AND TOTAL LOADING
   30 IF(N .EQ. NCASES)
     1      CALL STG3ES(DUM1,DUM2,DUM3,CLLP,NEWMAX,NOALFA,LCASE)
C
C  READ THE FIRST RUN SOLUTION
      CALL READMS(1,CPC,NEWMAX,IREAD)
C
C  DEFINE CP AND EP ARRAYS FOR THE PRESENT SECOND RUN FUNDAMENTAL CASE
      DO 40  I = 1,NEWMAX
      CPRC(I) = CP(I,LCASE)
      EP(I) = EPS(I,LCASE)
   40 CONTINUE
C
C  CALCULATE THE STABILITY DERIVATIVES
      CALL SUMIT1(CPC,CPA,CPP,CPRC,CPRA,DEL,EP,CHORD,Y,DELTA,CMU,XL,TL,
     1    XMC,AREA,DUM1,DUM2,CNP2,CLLRC,DUM3,DUM5,DUM6,CYP2,
     2    NW,IJ,NROWS,ISYMM,NEWMAX,IFUDGE)
      IF(N .LT. NCASES) GO TO 50
      CNPC = DUM1
      GO TO 110
   50 CALL SUMIT2(CPC,CPA,CPRC,CPRA,DEL,EP,CHORD,Y,DELTA,AREA,CMU,XL,TL,
     1      XMC,CNRC,DUM1,DUM2,CNR2C,DUM3,DUM4,CYRC,DUM5,DUM6,
     2      CYR2C,DUM7,DUM8,NW,IJ,NROWS,ISYMM,NEWMAX)
C
C  PRINT THE STABILITY DERIVATIVE DATA
   60 WRITE(6, 70 ) LCASE
   70 FORMAT(1H1,32X,13(4H****), 3H*** / 33X,
     1      49H*  STABILITY DERIVATIVE DATA FOR FUNDAMENTAL CASE, I3,
     2      3H  * / 33X,13(4H****), 3H*** /)
      WRITE(6, 80 ) CLLRC
   80 FORMAT(1H0/ 25X, 44HROLLING MOMENT COEFFICIENT DERIVATIVE DUE TO,
     1      15H YAWING, CLLR =, F12.7)
      WRITE(6, 90 ) CNRC,CNR2C
   90 FORMAT(1H0/12X,42HYAWING MOMENT COEFFICIENT ABOUT XMC DUE TO,
     1     53H YAWING ABOUT XCG, CN(R) MAY BE CALCULATED AS FOLLOWS //
     2      49X, 25HCN(R) = CNR*R + CNR2*R**2 //
     3      49X, 13HWHERE   CNR =,E14.9 / 56X, 6HCNR2 =,F12.7)
      WRITE(6, 100 ) CYRC,CYR2C
  100 FORMAT(1H0/ 23X,43HSIDE FORCE COEFFICIENT DUE TO YAWING, CY(R),
     1      29H MAY BE CALCULATED AS FOLLOWS//
     2      49X,25HCY(R) = CYR*R + CYR2*R**2//
     3      49X,13HWHERE   CYR =,E14.9/ 56X,6HCYR2 =,F12.7)
     4      44H ROLLING, CN(P) MAY BE CALCULATED AS FOLLOWS //
      GO TO 130
  110 WRITE(6, 70 ) LCASE
      WRITE(6, 120 ) CLLP,CNP2,CYP2
  120 FORMAT(1H0/ 28X, 33HROLLING MOMENT COEFF DERIVATIVE DUE TO,
     .      16H ROLLING, CLLP =, F12.7 ///
     2      17X, 42HYAWING MOMENT COEFFICIENT ABOUT XMC DUE TO,
     3      42HROLLING, CN(P) MAY BE CALCULATED AS FOLLOWS //
     4      53X, 17HCN(P) = CNP2*P**2 //
     5      53X, 13HWHERE   CNP2 =,F12.7///
     6      30X,44HSIDE FORCE COEFFICIENT DUE TO ROLLING, CY(P),
     7      29H MAY BE CALCULATED AS FOLLOWS//
     8      53X,17HCY(P) = CYP2*P**2//
     9      53X,13HWHERE   CYP2 =,F12.7)
  130 CONTINUE
      RETURN
      END
```

```
*DECK COMDER
      SUBROUTINE COMDER(EPS,CPC,CPA,CPRC,CPRA,CPP,CPREAD,DEL,CHORD,Y,
     1    CMU,DELTA,AREA,CLQ,CMQ,CMOMC,CLLP,CNP2,NW,IJ,NMAX,NJT,NEWMAX,
     2    NCASES,NROWS,ISYMM,XL,TL,XMC)
C
C  THIS SUBROUTINE CONTROLS CALCULATION OF STABILITY DERIVATIVES
C  FOR ALL COMPOSITE CASES
C
      COMMON/SOLV1/CP(600,10)
      COMMON/COMPOS/FACTOR(10,24),NCC
      DIMENSION CPC(NEWMAX),CPA(NEWMAX),CPRC(600),CPRA(600),CPP(600),
     1          CPREAD(NEWMAX)
      DIMENSION DEL(600),EPS(600,10),EP(600)
      DIMENSION CHORD(40),Y(40),DELTA(40),CMU(40),XL(40),TL(40)
      DIMENSION NW(40),IJ(40)
C
C  CYCLE THROUGH ALL COMPOSITE CASES
      IFUDGE = 1
      DO 100 M = 1,NCC
      MCASE = M
      NC1 = NCASES - 1
C  FIND THE SPOT WHERE THE SOLUTION OF THE FIRST RUN WAS STORED
      IREAD = NMAX + NJT + 1
C
C  DEFINE CP AND EP ARRAYS FOR THIS SECOND RUN COMPOSITE CASE
C  NOTE THAT CPP HAS PREVIOUSLY BEEN DEFINED IN FUNDER
      DO 20 I = 1,NEWMAX
      CPRC(I) = 0.00
      EP(I) = 0.00
      CPRA(I) = CP(I,1)
      DO 10 N = 1,NC1
      CPRC(I) = CPRC(I) + FACTOR(N,M) * CP(I,N)
      EP(I) = EP(I) + FACTOR(N,M) * EPS(I,N)
   10 CONTINUE
   20 CONTINUE
C
C  DEFINE CP FOR THE FIRST RUN COMPOSITE CASE
      IF(M .EQ. 1) CALL READMS(1,CPA,NEWMAX,IREAD)
      DO 30 I = 1,NEWMAX
      CPC(I) = 0.00
   30 CONTINUE
      IREAD = IREAD - 1
      DO 60 N = 1,NC1
      IREAD = IREAD + 1
      IF(FACTOR(N,M) .EQ. 0.0) GO TO 60
   40 CALL READMS(1,CPREAD,NEWMAX,IREAD)
      DO 50 I = 1,NEWMAX
      CPC(I) = CPC(I) + FACTOR(N,M) * CPREAD(I)
   50 CONTINUE
   60 CONTINUE
C  CALCULATE THE COMPOSITE CASE DERIVATIVES
   70 CALL SUMIT1(CPC,CPA,CPP,CPRC,CPRA,DEL,EP,CHORD,Y,DELTA,CMU,XL,TL,
     1      XMC,AREA,CNPC,CNPA,CNP2,CLLRC,CLLRA,CYPC,CYPA,CYP2,
     2      NW,IJ,NROWS,ISYMM,NEWMAX,IFUDGE)
   80 CALL SUMIT2(CPC,CPA,CPRC,CPRA,DEL,EP,CHORD,Y,DELTA,AREA,CMU,XL,TL,
     1 XMC,CNRC,CNRA,CNRA2,CNR2C,CNR2A,CNR2A2,CYRC,CYRA,CYRA2,
     2 CYR2C,CYR2A,CYR2A2,NW,IJ,NROWS,ISYMM,NEWMAX)
C
C  PRINT A SUMMARY TABLE OF ALL STABILITY DERIVATIVES
   90 CALL STABLE(CLQ,CMQ,CMOMC,CLLP,CNPC,CNPA,CNP2,CYPC,CYPA,CYP2,
     1      CLLRC,CLLRA,CNRC,CNRA,CNRA2,CNR2C,CNR2A,CNR2A2,
     2      CYRC,CYRA,CYRA2,CYR2C,CYR2A,CYR2A2,MCASE)
  100 CONTINUE
      RETURN
      END
```

```
*DECK SUMIT1
      SUBROUTINE SUMIT1(CPO,CPA,CPP,CPRO,CPRA,DEL,EPS,CHORD,Y,DELTA,CMU,
     1  XL,TL,XMC,AREA,CNPO,CNPA,CNP2,CLLRO,CLLRA,CYPO,CYPA,CYP2,
     2  NW,IJ,NROWS,ISYMM,NEWMAX,IFUDGE)
C
C  THIS SUBROUTINE INTEGRATES CP CHORDWISE AND SPANWISE TO CALCULATE
C  THE TERMS OF CNP AND CLLR DERIVATIVES
C
      COMMON /DERIV/ UO(40),CLO,CMO,CMOMC
      DIMENSION CPO(NEWMAX),CPA(NEWMAX),CPP(600),CPRO(600),CPRA(600)
      DIMENSION DEL(600),EPS(600)
      DIMENSION CHORD(40),Y(40),DELTA(40),CMU(40),XL(40),TL(40)
      DIMENSION NW(40),IJ(40)
C
C  INITIALIZE THE DERIVATIVE TERMS
   10 CNPO = 0.00
      CNPA = 0.00
      CNP2 = 0.00
      CYPO = 0.00
      CYPA = 0.00
      CYP2 = 0.00
      CLLRO = 0.00
      CLLRA = 0.00
C
C  INTEGRATE THE COEFFICIENT TERMS SPANWISE
      I = 0
      DO 100 K = 1,NROWS
C
C  INTEGRATE THE COEFFICIENT TERMS CHORDWISE
C  LEADING EDGE CONTRIBUTION
      I = I + 1
C  YAWING DUE TO ROLLING
   20 TERMP = DEL(I) * (CPP(I) + 0.50*CPP(I+1))
      EP = -EPS(I) / (57.295779 * UO(K))
      DPO = TERMP * EP
      DPA = TERMP
      SPO = 0.3490658 * DEL(I) * CPP(I) * CPO(I)
      SPA = 0.3490658 * DEL(I) * CPP(I) * CPA(I)
      SP2 = 0.1745329 * DEL(I) * CPP(I)**2
C  ROLLING DUE TO YAWING
   30 CLGO = DEL(I) * (CPRO(I) + 0.50*CPRO(I+1))
      CLGA = DEL(I) * (CPRA(I) + 0.50*CPRA(I+1))
C
C  REGULAR EVD CONTRIBUTIONS
   40 NWK = NW(K)
      DO 70  L = 2,NWK
      I = I + 1
      CPP1 = CPP(I+1)
      CPRO1 = CPRO(I+1)
      CPRA1 = CPRA(I+1)
      IF(L .LT. NWK) GO TO 50
      CPP1 = 0.00
      CPRO1 = 0.00
      CPRA1 = 0.00
      IF(CMU(K) .LT. 0.0001) GO TO 50
      IJK = IJ(K)
      CPP1 = CPP(IJK)
      CPRO1 = CPRO(IJK)
      CPRA1 = CPRA(IJK)
   50 EP = -EPS(I) / (57.295779 * UO(K))
      TERMP = 0.50 * DEL(I) * (CPP(I) + CPP1)
      DPO = DPO + TERMP * EP
      DPA = DPA + TERMP
   60 CLGO = CLGO + 0.50 * DEL(I) * (CPRO(I) + CPRO1)
      CLGA = CLGA + 0.50 * DEL(I) * (CPRA(I) + CPRA1)
   70 CONTINUE
C
C  INTEGRATE THE COEFFICIENT TERMS SPANWISE
      FACTOR = CHORD(K) * Y(K) * DELTA(K)
      FACTO = CHORD(K) * DELTA(K) * TL(K)
      FACT = FACTO * (XL(K)-XMC)
   80 CNPO = CNPO + (DPO - SPO) * FACTOR - SPO * FACT
      CNPA = CNPA + (DPA/57.295779 - SPA) * FACTOR - SPA * FACT
      CNP2 = CNP2 + SP2 * FACTOR + SP2 * FACT
      CYPO = CYPO + SPO * FACTO
      CYPA = CYPA + SPA * FACTO
      CYP2 = CYP2 + SP2 * FACTO
   90 CLLRO = CLLRO + CLGO * FACTOR
      CLLRA = CLLRA + CLGA * FACTOR
  100 CONTINUE
C
C  PUT THE TERMS IN FINAL FORM
      FACTOR = 1.00 / AREA
      IF(ISYMM .LT. 0) FACTOR = 2.00 / AREA
  110 CNPO = FACTOR * CNPO
      CNPA = FACTOR * CNPA
      CNP2 = -FACTOR * CNP2
      CLLRO = -FACTOR * CLLRO
      CLLRA = -FACTOR * CLLRA
      FACTOR = 2.00 * FACTOR
      CYPO = FACTOR * CYPO
      CYPA = FACTOR * CYPA
      CYP2 = FACTOR * CYP2
      IF(IFUDGE .NE. 0) GO TO 120
      CNPO = 0.00
      CYPO = 0.00
  120 IF(ISYMM .GE. 0) GO TO 130
      CNP2 = 0.00
      CYP2 = 0.00
  130 RETURN
      END
```

```
*DECK SUMIT2
      SUBROUTINE SUMIT2(CPO,CPA,CPRO,CPRA,DEL,EPS,CHORD,Y,DELTA,AREA,
     1  CMU,XL,TL,XMC,CNRO,CNRA,CNRA2,CNR2O,CNR2A,CNR2A2,CYRO,CYRA,
     2  CYRA2,CYR2O,CYR2A,CYR2A2,NW,IJ,NROWS,ISYMM,NEWMAX)
C
C  THIS SUBROUTINE INTEGRATES CP CHORDWISE AND SPANWISE TO CALCULATE
C  THE TERMS OF THE YAWING AND SIDE FORCE COEFFICIENTS DUE TO YAWING
C
      COMMON /DERIV/ UO(40),CLO,CMO,CMOMC
      DIMENSION CPO(NEWMAX),CPA(NEWMAX),CPRO(600),CPRA(600)
      DIMENSION DEL(600),EPS(600)
      DIMENSION CHORD(40),Y(40),DELTA(40),CMU(40),XL(40),TL(40)
      DIMENSION NW(40),IJ(40)
C
C  INITIALIZE THE YAWING COEFFICIENT TERMS
   10 CNRO = 0.00
      CNRA = 0.00
      CNRA2 = 0.00
      CNR2O = 0.00
      CNR2A = 0.00
      CNR2A2 = 0.00
      CYRO = 0.00
      CYRA = 0.00
      CYRA2 = 0.00
      CYR2O = 0.00
      CYR2A = 0.00
      CYR2A2 = 0.00
C
      I = 0
      DO 100 K = 1,NROWS
C
C  INTEGRATE THE COEFFICIENT TERMS CHORDWISE
C  LEADING EDGE CONTRIBUTIONS
      I = I + 1
   20 TERMO = DEL(I) * (CPRO(I) + 0.50*CPRO(I+1))
      TERMA = DEL(I) * (CPRA(I) + 0.50*CPRA(I+1))
      EP = -EPS(I) / (57.295779 * UO(K))
      DRO = TERMO * EP
      DRA = TERMO/57.295779 + TERMA * EP
      DRA2 = TERMA
   30 SRO = 0.3490658 * DEL(I) * CPRO(I) * CPO(I)
      SRA = 0.3490658 * DEL(I) * (CPRO(I)*CPA(I) + CPRA(I)*CPO(I))
      SRA2 = 0.3490658 * DEL(I) * (CPRA(I)*CPA(I))
      IF(ISYMM .LT. 0) GO TO 40
      SR2O = 0.1745329 * DEL(I) * CPRO(I)**2
      SR2A = 0.3490658 * DEL(I) * CPRO(I)*CPRA(I)
      SR2A2 = 0.1745329 * DEL(I) * CPRA(I)**2
C
C  REGULAR EVD CONTRIBUTIONS
   40 NWK = NW(K)
      DO 70  L = 2,NWK
      I = I + 1
      CPRO1 = CPRO(I+1)
      CPRA1 = CPRA(I+1)
      IF(L .LT. NWK) GO TO 50
      CPRO1 = 0.00
      CPRA1 = 0.00
      IF(CMU(K) .LT. 0.0001) GO TO 50
      IJK = IJ(K)
      CPRO1 = CPRO(IJK)
      CPRA1 = CPRA(IJK)
   50 EP = -EPS(I) / (57.295779 * UO(K))
C  REGULAR EVD CONTRIBUTIONS
      TERMO = 0.50 * DEL(I) * (CPRO(I)+CPRO1)
   60 TERMA = 0.50 * DEL(I) * (CPRA(I)+CPRA1)
      DRO = DRO + TERMO * EP
      DRA = DRA + TERMO/57.295779 + TERMA * EP
      DRA2 = DRA2 + TERMA
   70 CONTINUE
C
C  INTEGRATE THE COEFFICIENT TERMS SPANWISE
      FACTOR = CHORD(K) * Y(K) * DELTA(K)
      FACTO = CHORD(K) * TL(K) * DELTA(K)
      FACT = FACTO * (XL(K)-XMC)
   80 CNRO = CNRO + (DRO-SRO) * FACTOR - SRO * FACT
      CNRA = CNRA + (DRA-SRA) * FACTOR - SRA * FACT
      CNRA2 = CNRA2 + (DRA2/57.295779-SRA2) * FACTOR - SRA2 * FACT
      CYRO = CYRO + SRO * FACTO
      CYRA = CYRA + SRA * FACTO
      CYRA2 = CYRA2 + SRA2 * FACTO
      IF(ISYMM .LT. 0) GO TO 100
   90 CNR2O = CNR2O - SR2O * FACTOR -SR2O * FACT
      CNR2A = CNR2A - SR2A * FACTOR - SR2A * FACT
      CNR2A2 = CNR2A2 - SR2A2 * FACTOR - SR2A2 * FACT
      CYR2O = CYR2O + SR2O * FACTO
      CYR2A = CYR2A + SR2A * FACTO
      CYR2A2 = CYR2A2 + SR2A2 * FACTO
  100 CONTINUE
C
C  PUT THE TERMS IN FINAL FORM
      FACTOR = 1.00 / AREA
      IF(ISYMM .LT. 0) FACTOR = 2.00 / AREA
  110 CNRO = FACTOR * CNRO
      CNRA = FACTOR * CNRA
      CNRA2 = FACTOR * CNRA2
      CNR2O = FACTOR * CNR2O
      CNR2A = FACTOR * CNR2A
      CNR2A2 = FACTOR * CNR2A2
      FACTOR = 2.00 * FACTOR
      CYRO = FACTOR * CYRO
      CYRA = FACTOR * CYRA
      CYRA2 = FACTOR * CYRA2
      CYR2O = FACTOR * CYR2O
      CYR2A = FACTOR * CYR2A
      CYR2A2 = FACTOR * CYR2A2
  120 RETURN
      END
```

```
*DECK STABLE
      SUBROUTINE STABLE(CLQ,CMQ,CMQMC,CLLP,CNPO,CNPA,CNP2,CYPO,CYPA,
     1       CYP2,CLLRO,CLLRA,CNRO,CNRA,CNRA2,CNR20,CNR2A,CNR2A2,
     2       CYRO,CYRA,CYRA2,CYR20,CYR2A,CYR2A2,MCASE)
C
C   THIS SUBROUTINE CALCULATES AND PRINTS A COMPLETE SUMMARY TABLE
C   OF ALL STABILITY DERIVATIVE DATA FOR EACH COMPOSITE CASE
C
      COMMON/COMPOS/FACTOR(10,24),NCC
C
C   PRINT ALL CONSTANT DERIVATIVES
      WRITE(6, 10 ) MCASE
   10 FORMAT(1H1,33X,13(4H****),1H* /  34X,
     1       47H*  STABILITY DERIVATIVE DATA FOR COMPOSITE CASE, I3,
     2       3H  */ 34X,13(4H****),1H* /)
      WRITE(6, 20 ) (N,N=1,10), (FACTOR(N,MCASE),N=1,10)
   20 FORMAT(1H ,48X,24HFUNDAMENTAL CASE FACTORS/ 10X,9(4X,2HA(,I1,1H),
     1     2X),3X,2HA(,I2,1H) / 10X,10F10.6)
      WRITE(6, 30 ) CLQ,CMQ,CMQMC
   30 FORMAT(1H0/ 26X,43HLIFT COEFFICIENT DERIVATIVE DUE TO PITCHING,
     1       17H ABOUT XCG, CLQ =,F10.6 /
     2       14X,51HPITCHING MOMENT COEFFICIENT DERIVATIVE ABOUT ORIGIN,
     3       33H DUE TO PITCHING ABOUT XCG, CMQ =,F10.6 /
     4       16X,42HPITCHING MOMENT COEFF DERIVATIVE ABOUT XMC,
     5       35H DUE TO PITCHING ABOUT XCG, CMQMC =, F10.6)
      WRITE(6, 40 ) CLLP,CNPO,CNPA,CNP2
   40 FORMAT(1H0/28X,38HROLLING MOMENT COEFF DERIVATIVE DUE TO,
     1       16H ROLLING, CLLP =,F12.7 ///
     2       17X,51HYAWING MOMENT COEFFICIENT ABOUT XMC DUE TO ROLLING,,
     3       35H CN(P) MAY BE CALCULATED AS FOLLOWS /
     4       48X,25HCN(P) = CNP*P + CNP2*P**2 /
     5       42X,31HWHERE    CNP = CNPO + CNPA*ALPHA /
     6       52X, 6HCNPO =,F12.7/52X, 6HCNPA =,F12.7/52X,6HCNP2 =,F12.7)
      WRITE(6, 50 ) CYPO,CYPA,CYP2
   50 FORMAT(1H0/26X,44HSIDE FORCE COEFFICIENT DUE TO ROLLING, CY(P),
     1       29H MAY BE CALCULATED AS FOLLOWS/
     2       48X,25HCY(P) = CYP*P + CYP2*P**2/
     3       42X,31HWHERE    CYP = CYPO + CYPA*ALPHA/
     4       52X, 6HCYPO =,F12.7/52X,6HCYPA =,F12.7/52X,6HCNP2 =,F12.7)
      WRITE(6, 60 ) CLLRO,CLLRA
   60 FORMAT(1H0/15X,39HROLLING MOMENT COEFF DERIVATIVE DUE TO,
     1       52H YAWING ABOUT XCG, CLLR MAY BE CALCULATED AS FOLLOWS /
     2       48X,26HCLLR = CLLRO + CLLRA*ALPHA /
     3       48X,15HWHERE    CLLRO =,F13.7 / 56X,7HCLLRA =,F13.7)
      WRITE(6, 70 ) CNRO,CNRA,CNRA2,CNR20,CNR2A,CNR2A2
   70 FORMAT(1H0/12X,42HYAWING MOMENT COEFFICIENT ABOUT XMC DUE TO,
     1       53H YAWING ABOUT XCG, CN(R) MAY BE CALCULATED AS FOLLOWS /
     2       48X,25HCN(R) = CNR*R + CNR2*R**2 /
     3       36X,48HWHERE    CNR = CNRO + CNRA*ALPHA + CNRA2*ALPHA**2 /
     4       52X,6HCNRO =,F13.7/52X,6HCNRA =,F13.7/51X,7HCNRA2 =,F13.7/
     5     36X,51HAND     CNR2 = CNR20 + CNR2A*ALPHA + CNR2A2*ALPHA**2/
     6   51X,7HCNR20 =,F13.7/51X,7HCNR2A =,F13.7/50X,8HCNR2A2 =,F13.7)
      WRITE(6, 80 ) CYRO,CYRA,CYRA2,CYR20,CYR2A,CYR2A2
   80 FORMAT(1H0/13X,39HSIDE FORCE COEFFICIENT ABOUT XMC DUE TO,
     1       53H YAWING ABOUT XCG, CY(R) MAY BE CALCULATED AS FOLLOWS /
     2       48X,25HCY(R) = CYR*R + CYR2*R**2 /
     3       36X,48HWHERE    CYR = CYRO + CYRA*ALPHA + CYRA2*ALPHA**2 /
     4       52X,6HCYRO =,F13.7/52X,6HCYRA =,F13.7/51X,7HCYRA2 =,F13.7/
     5     36X,51HAND     CYR2 = CYR20 + CYR2A*ALPHA + CYR2A2*ALPHA**2/
     6   51X,7HCYR20 =,F13.7/51X,7HCYR2A =,F13.7/50X,8HCYR2A2 =,F13.7)
C
C   PRINT TABLE OF DERIVATIVE TERMS WHICH DEPEND ON ALPHA
      WRITE(6, 90 )
   90 FORMAT(1H1,32X,3H***,10(5H*****),/,
     1 33X,53H* VARIATION OF STABILITY TERMS WITH ANGLE OF ATTACK *,/,
     2 33X,3H***,10(5H*****),//,
     3 9X,5HALPHA,12X,3HCNP,10X,4HCNP2,14X,3HCYP,10X,4HCYP2,
     4 14X,4HCLLR, 9X,3HCNR,10X,4HCNR2)
      ALPHA = -11.00
      DO 140 M = 1,41
  100 ALPHA = ALPHA + 1.00
      CNP = CNPO + CNPA * ALPHA
       CYP = CYPO + CYPA * ALPHA
  110 CLLR = CLLRO + CLLRA * ALPHA
      CNR = CNRO + CNRA * ALPHA + CNRA2 * ALPHA**2
  120 CNR2 = CNR20 + CNR2A * ALPHA + CNR2A2 * ALPHA**2
      WRITE(6, 130 ) ALPHA,CNP,CNP2,CYP,CYP2,CLLR,CNR,CNR2
  130 FORMAT(1H , 5X,F10.6,5H   *  ,2F13.7,5H   *  ,2F13.7,5H   *
     1          3F13.7)
  140 CONTINUE
      RETURN
      END
```

```
*DECK STAGE4
      OVERLAY(OVLY,4,0)
      PROGRAM STAGE4
C
C   THIS PROGRAM CONTROLS THE EXECUTION OF UTILITY ROUTINES AND
C   BOUNDARY CONDITION SETUP FOR STABILITY DERIVATIVE RUNS
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1              D(40),KK(600),ITYPE(600)
      COMMON/FCASE2/TST(40,10),HL(40,10),DJ(40),ACTE(40),AC(20,40),
     1              XHB(4,40),BET(4,40),IFS(4,40),ICT(40),IHT(40),NCT,NHT
      COMMON/FCASE3/EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10)
      COMMON/SOLV1/CP(600,10)
      DIMENSION CPREAD(600)
C
C   ADD THE EXTRA FUNDAMENTAL CASE FOR DERIVATIVES DUE TO PITCHING
      IF(LOGIC .GT. 1) GO TO 60
   10 IF(NCASES .LT. 10) GO TO 30
      WRITE(6, 20 )
   20 FORMAT(1H0//14X,44HFUNDAMENTAL CASE 10 HAS BEEN REPLACED BY THE,
     1       47H FOLLOWING CASE FOR DERIVATIVES DUE TO PITCHING)
      GO TO 40
   30 NCASES = NCASES + 1
   40 CALL BCPICH(XCG,CREF,XI,DEL,EPS,BETA,CHORD,KK,THETA,THS,TST,HL,
     1       NW,IW,NJ,IJ,NWT,NMAX,NROWS,NCASES)
   50 CALL OUT2(NCASES)
      RETURN
C
C   SAVE THE FIRST RUN SOLUTION ON UNIT 1
   60 ISIZE = NEWMAX
      CALL SAVECP(CP,CPREAD,NMAX,NJT,ISIZE,NCASES)
C
C   DEFINE THE FUNDAMENTAL CASES FOR YAWING DERIVATIVES
      NC1 = NCASES - 1
   70 CALL BCYAW(EPS,BETA,THETA,THS,Y,KK,NWT,NMAX,NROWS,NC1)
C   DEFINE THE LAST FUNDAMENTAL CASE FOR ROLLING DERIVATIVES
   80 CALL BCROLL(EPS,BETA,THETA,THS,TST,Y,NW,NWT,NMAX,NROWS,NCASES)
C
C   PRINT THE FUNDAMENTAL CASE GEOMETRY
      IF(IPRINT .GE. 0) GO TO 100
      DO 90  N = 1,NCASES
      LCASE = N
      CALL OUT2(LCASE)
   90 CONTINUE
  100 CONTINUE
      END
```

```
*DECK OUT2
      SUBROUTINE OUT2(LCASE)
C
C   THIS SUBROUTINE PRINTS OUT THE GEOMETRIC DATA DERIVED FROM THE
C   SECTIONAL METHOD INPUT
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON /LUKE/ TITLE(8)
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1              D(40),KK(600),ITYPE(600)
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)
      COMMON/FCASE2/TST(40,10),HL(40,10),DJ(40),ACTE(40),AC(20,40),
     1              XHB(4,40),BET(4,40),IFS(4,40),ICT(40),IHT(40),NCT,NHT
      COMMON/FCASE3/EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10)
      COMMON/INDATA/ARE,SPA,CRE,XM,CMA,XC,NRO,NC,ISY,IPR,JET,IGT,IHI
C
C   PRINT CASE TITLE AND GENERAL GEOMETRIC PARAMETERS
      IF(LCASE .GT. 1) GO TO 60
   10 WRITE(6, 20 )TITLE
   20 FORMAT(1H1,39X,10(4H****)/
     1    40X,40H*   EVD   JET - WING   COMPUTER   PROGRAM   */
     2    40X,10(4H****)///20X,8A10)
      CMA = CMAC * SPA / 2.0
   30 WRITE(6, 40 ) AREA,ARE,SPAN,SPA,CREF,CRE,XMC,XM,CMAC,CMA,ARATIO,
     1              ARATIO,XCG,XC
   40 FORMAT(1H0//54X,4HUSED,11X,5HINPUT /
     1       41X,6HAREA =,2F15.6 / 41X,6HSPAN =,2F15.6 /
     2       41X,6HCREF =,2F15.6 / 42X,5HXMC =,2F15.6 /
     3       41X,6HCMAC =,2F15.6 / 39X,8HARATIO =,2F15.6 /
     4       42X,5HXCG =,2F15.6)
      WRITE(6, 50 ) NROWS,NRO,NCASES,NC,ISYMM,ISY,IPRINT,IPR,JETFLG,JET,
     1              IGTYPE,IGT,IHINGE,IHI,NWT,NJT,NMAX
   50 FORMAT(1H0/ 48X,7HNROWS =,I3,7X,I3 / 47X,8HNCASES =,I3,7X,I3 /
     1              48X,7HISYMM =,I3,7X,I3 / 47X,8HIPRINT =,I3,7X,I3 /
     2              47X,8HJETFLG =,I3,7X,I3 / 47X,8HIGTYPE =,I3,7X,I3 /
     3              47X,8HIHINGE =,I3,7X,I3 ///
     4              43X,25HNUMBER OF WING ELEMENTS =,I4 /
     5              43X,25HNUMBER OF JET ELEMENTS  =,I4 /
     6              42X,26HTOTAL NUMBER OF ELEMENTS =,I4)
   60 J = 0
      JJ = NWT
C
C   PRINT FUNDAMENTAL CASE HEADER
      WRITE(6, 70 ) LCASE
   70 FORMAT(1H1,23X,1H*,19(4H****)/
     1    24X,54H*  ELEMENT GEOMETRY DATA AND FUNDAMENTAL CASE DATA FOR,
     2    17H FUNDAMENTAL CASE,I2,3H  */24X,1H*,19(4H****))
      ILINES = 3
      DO 260 K = 1,NROWS
C
C   PRINT SECTIONAL DATA
      WRITE(6, 80 ) K,Y(K),DELTA(K),XLEAD(K),XTRAIL(K),CHORD(K),TANLE(K)
```

```
   80 FORMAT(1H0,11H*** SECTION,I3,4H ***,2X,3HY =,F10.6,2X,7HDELTA =,
     1 F10.6,2X,7HXLEAD =,F10.6,2X,8HXTRAIL =,F10.6,2X,7HCHORD =,F10.6,
     2 2X,7HTANLE =,F10.6)
C
C  PRINT CHORDWISE DATA ON WING
      NWK = NW(K)
      WRITE(6, 90 ) NWK,TST(K,LCASE),HL(K,LCASE),THS(K,LCASE)
   90 FORMAT(1H0,21HWING ELEMENTS     NW =,I3,5X, 7HTWIST =,F10.6,5X,
     1           4HHL =,F10.6,5X, 9HTHETA S =,F10.6)
      WRITE(6, 100 ) (XB(J+L),L=1,NWK)
  100 FORMAT(1H ,14X,2HXB,  10F11.6/17X,10F11.6)
      IF(LCASE .GT. 1) GO TO 130
      WRITE(6, 110 ) (XI(J+L),L=1,NWK)
  110 FORMAT(1H ,14X, 2HXI,10F11.6/17X,10F11.6)
      WRITE(6, 120 ) (DEL(J+L),L=1,NWK)
  120 FORMAT(1H ,13X,3HDEL,10F11.6/17X,10F11.6)
  130 IF(ICT(K) .EQ. 0) GO TO 150
      ICK = ICT(K)
      WRITE(6, 140 ) (AC(L,ICK),L=1,NWK)
  140 FORMAT(1H ,10X,6HCAMBER,10F11.6/17X,10F11.6)
  150 WRITE(6, 160 ) (EPS(J+L,LCASE),L=1,NWK)
  160 FORMAT(1H ,13X,3HEPS,10F11.6/17X,10F11.6)
      WRITE(6, 170 ) (BETA(J+L,LCASE),L=1,NWK)
  170 FORMAT(1H ,12X,4HBETA,10F11.6/17X,10F11.6)
      WRITE(6, 180 ) (ITYPE(J+L),L=1,NWK)
  180 FORMAT(1H ,12X,4HTYPE,10(3X,I2,6X)/17X,10(3X,I2,6X))
      J = J + NWK
      IL = 1
      IF(NWK .GT. 9) IL = 2
      ILINES = ILINES + 4 + 4*IL
      IF(LCASE .EQ. 1) ILINES = ILINES + 2*IL
C
C  PRINT CHORDWISE DATA ON JET
      NJK = NJ(K)
      IF(NJK .GT. 0) GO TO 200
      WRITE(6, 190 )
  190 FORMAT(1H ,8X,19HTHIS ROW HAS NO JET)
      ILINES = ILINES + 1
      GO TO 230
  200 WRITE(6, 210 ) NJK,D(K),DJ(K),ACTE(K),THETA(K,LCASE)
  210 FORMAT(1H0,1X,20HJET ELEMENTS     NJ =,I3,5X,3HD =,F10.6,5X,4HDJ =,
     1    F10.6,5X,6HACTE =,F10.6,5X,7HTHETA =,F10.6)
      WRITE(6, 100 ) (XB(JJ+L),L=1,NJK)
      IF(LCASE .GT. 1) GO TO 220
      WRITE(6, 110 ) (XI(JJ+L),L=1,NJK)
      WRITE(6, 120 ) (DEL(JJ+L),L=1,NJK)
  220 WRITE(6, 170 ) (BETA(JJ+L,LCASE),L=1,NJK)
      WRITE(6, 180 ) (ITYPE(JJ+L),L=1,NJK)
      JJ = JJ + NJK
      IL = 1
      IF(NJK .EQ. 10) IL = 2
      ILINES = ILINES + 1 + 3 * IL
      IF(LCASE .EQ. 1) ILINES = ILINES + 2*IL
  230 IF(K .EQ. NROWS) GO TO 260
      NWK1 = NW(K+1)
      IL = 1
      IF(NWK1 .GT. 9) IL = 2
      NEXT = 4 + 4*IL
      IF(LCASE .EQ. 1) NEXT = NEXT + 2*IL
      NJK1 = NJ(K+1)
      IL = 1
      IF(NJK1 .EQ. 10) IL = 2
      NEXT = NEXT + 1
      IF(NJK1 .EQ. 0) GO TO 240
      NEXT = NEXT + 1 + 3*IL
      IF(LCASE .EQ. 1) NEXT = NEXT + 2*IL
  240 IF((55-ILINES) .GE. NEXT) GO TO 260
      WRITE(6, 250 )
  250 FORMAT(1H1)
      ILINES = 1
  260 CONTINUE
      RETURN
      END
```

```
*DECK SAVECP
      SUBROUTINE SAVECP(CP,DUMMY,NMAX,NJT,ISIZE,NCASES)
C
C  THIS SUBROUTINE SAVES THE CP SOLUTION FOR ALL FUNDAMENTAL CASES
C  OF THE FIRST STABILITY DERIVATIVE RUN BY STORING ON DIRECT ACCESS
C
      DIMENSION CP(600,10),DUMMY(ISIZE)
C
C  FIND THE PROPER PLACE TO WRITE THE OLD SOLUTION
   10 IWRITE = NMAX + NJT
C  DEFINE THE DUMMY ARRAY
      DO 30 N = 1,NCASES
      IWRITE = IWRITE + 1
      DO 20 I = 1,ISIZE
      DUMMY(I) = CP(I,N)
   20 CONTINUE
C
C  SAVE THE DATA
      CALL WRITMS(1,DUMMY,ISIZE,IWRITE)
   30 CONTINUE
      RETURN
      END
```

```
*DECK BCPICH
      SUBROUTINE BCPICH(XCG,CREF,XI,DEL,EPS,BETA,C,KK,THETA,THS,TST,HL,
     1                  NW,IW,NJ,IJ,NWT,NMAX,NROWS,N)
C
C  THIS SUBROUTINE DEFINES THE BOUNDARY CONDITIONS FOR
C  THE PITCHING RATE DERIVATIVE FUNDAMENTAL CASE
C
      DIMENSION XI(600),DEL(600),C(40),KK(600)
      DIMENSION NW(40),IW(40),NJ(40),IJ(40)
      DIMENSION EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10)
      DIMENSION TST(40,10),HL(40,10)
C
C  DEFINE THE CAMBER ANGLES WHICH RESULT FROM PITCHING
      DO 20 I = 1,NWT
      KKI = KK(I)
   10 EPS(I,N) = 2.00 * (XI(I)+(DEL(I)/2.0)*C(KKI)-XCG) / CREF
      BETA(I,N) = 0.00
   20 CONTINUE
      IF(NWT .EQ. NMAX) GO TO 50
      NWT1 = NWT + 1
      DO 40 I = NWT1,NMAX
   30 EPS(I,N) = 0.00
      BETA(I,N) = 0.00
   40 CONTINUE
C
C  DEFINE THE JET ANGLES WHICH RESULT FROM PITCHING
   50 DO 70 K = 1,NROWS
   60 THETA(K,N) = 0.00
      IJK = IJ(K)
      IF(NJ(K) .GT. 0) THETA(K,N) = 2.00 * (XI(IJK) - XCG) / CREF
      THS(K,N) = 0.00
      TST(K,N) = 0.00
      HL(K,N) = 0.00
   70 CONTINUE
      RETURN
      END
```

```
*DECK BCROLL
      SUBROUTINE BCROLL(EPS,BETA,THETA,THS,TST,Y,
     1                  NW,NWT,NMAX,NROWS,NCASES)
C
C  THIS SUBROUTINE DEFINES THE BOUNDARY CONDITIONS FOR
C  THE ROLLING RATE DERIVATIVE FUNDAMENTAL CASE
C
      DIMENSION EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10),
     1          TST(40,10)
      DIMENSION Y(40),NW(40)
C
C  DEFINE THE TWIST AND CAMBER ANFLES WHICH RESULT FROM ROLLING
      N = NCASES
      I = 0
      DO 40 K = 1,NROWS
   10 TST(K,N) = Y(K)
      THETA(K,N) = TST(K,N)
      THS(K,N) = 0.00
      NWK = NW(K)
      DO 30 L = 1,NWK
      I = I + 1
   20 EPS(I,N) = TST(K,N)
   30 CONTINUE
   40 CONTINUE
C
C  DEFINE THE ANGLES ON THE JET
      IF(NMAX .EQ. NWT) RETURN
      NWT1 = NWT + 1
      DO 60 I = NWT1,NMAX
   50 EPS(I,N) = 0.00
   60 CONTINUE
      RETURN
      END
```

```
*DECK BCYAW
      SUBROUTINE BCYAW (EPS,BETA,THETA,THS,Y,KK,
     1                  NWT,NMAX,NROWS,NCASES)
C
C  THIS SUBROUTINE DEFINES THE BOUNDARY CONDITIONS FOR ALL OF THE
C  YAWING RATE DERIVATIVE FUNDAMENTAL CASES
C
      COMMON /DERIV/ UO(40),CLO,CMO,CMOMC
      DIMENSION EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10)
      DIMENSION Y(40),KK(600)
C
C  DEFINE THE SECTIONAL NORMALIZED VELOCITY INDUCED BY YAWING
      DO 10 K = 1,NROWS
      UO(K) = Y(K) / 57.295779
   10 CONTINUE
C
C  DEFINE THE ANGLES FOR ALL FUNDAMENTAL CASES
      DO 80  N = 1,NCASES
      DO 30 I = 1,NWT
      KKI = KK(I)
   20 EPS(I,N) = -UO(KKI) * EPS(I,N)
      BETA(I,N) = 0.00
   30 CONTINUE
C
      DO 50 K = 1,NROWS
   40 THETA(K,N) = -UO(K) * THETA(K,N)
      THS(K,N) = 0.00
   50 CONTINUE
      IF(NMAX .EQ. NWT) GO TO 80
      NWT1 = NWT + 1
      DO 70 I = NWT1,NMAX
   60 EPS(I,N) = 0.00
      BETA(I,N) = 0.00
   70 CONTINUE
   80 CONTINUE
      RETURN
      END
```

TITLE CARD

TITLE
1–80

BASIC PLANFORM PARAMETER CARD

WING AREA	WING SPAN	CREF	XMC	XCG	
1	11	21	31	41	51–80

GENERAL CONTROL CARD

NROWS	NCASES	ISYMM	IPRINT	JETFLG	IGTYPE	IHINGE	IDERIV	
1					11			21–80

SECTION CENTERLINE LOCATIONS

$Y_{1,9,17,25,33}$	$Y_{2,10,18,26,34}$	$Y_{3,...}$	$Y_{4,...}$	$Y_{5,...}$	$Y_{6,...}$	$Y_{7,...}$	$Y_{8,...}$
1	11	21	31	41	51	61	71–80

WING SECTION TYPE CARD

ICTYPE $_{SECTION\ 1}$ .... ICTYPE $_{SECTION\ NROWS}$	HIGHEST VALUE INPUT = NWTYPE

Columns: 1, 11, 21, 31, 41, 51, 61, 71, 80

NUMBER OF CHORDWISE WING ELEMENTS

NI $_{SECTION\ TYPE\ 1}$ .... NI $_{SECTION\ TYPE\ NWTYPE}$	
1–20	21–80 (shaded)

WING CHORDWISE ELEMENT COORDINATES

$X_{1,9,17}$	$X_{2,10,18}$	$X_{3,11,19}$	$X_{4,12,20}$	$X_{5,13}$	$X_{6,14}$	$X_{7,15}$	$X_{8,16}$
1	11	21	31	41	51	61	71–80

AS MANY AS 10 SETS PERMITTED (1 < NWTYPE < 10)

LEADING AND TRAILING EDGE COORDINATES

Y	X_{LEAD}	X_{TRAIL}	ØNLY REQUIRED FØR SECTIØNS ADJACENT TØ LEADING ØR TRAILING EDGE BREAKS (THIS INPUT REQUIRED ØNLY IF IGTYPE = 1)
1	11	21	31 41 51 61 71 80
9			

AS MANY AS NRØWS CARDS PERMITTED (NRØWS ≤ 40)

TRAPEZOIDAL WING PARAMETERS

ASPECT RATIØ	SWEEP	TAPER RATIØ	(ØNLY REQUIRED IF IGTYPE = 2)
1	11	21	31 41 51 61 71 80

JET SECTION TYPE CARD

1	11	21	31	41	51	61	71–80
				$IJTYPE_{SECTION\ 1}$ .... $IJTYPE_{SECTION\ NROWS}$			HIGHEST VALUE INPUT = NJTYPE NUMBER ØF NØNZERØ VALUES = NRØWSJ

NUMBER OF CHORDWISE JET ELEMENTS

1	11	21–80
$NI_{SECTION\ TYPE\ 1}$ .... $NI_{SECTION\ TYPE\ NJTYPE}$		(unused)

JET CHORDWISE ELEMENT COORDINATES

1	11	21	31	41	51	61	71–80
$\bar{x}_{1,9,17}$	$\bar{x}_{2,10,18}$	$\bar{x}_{3,11,19}$	$\bar{x}_{4,12,20}$	$\bar{x}_{5,13}$	$\bar{x}_{6,14}$	$\bar{x}_{7,15}$	$\bar{x}_{8,16}$

AS MANY AS 10 SETS PERMITTED (NJTYPE ≤ 10)

FUNDAMENTAL CASE CONTROL CARD

INTWST	INHITE	INDELJ	INCAMB	INBETA								
1					11	21	31	41	51	61	71	80

TWIST DISTRIBUTION CARDS

$TWIST_{1,9,...}$	$TWIST_{2,10,...}$	$TWIST_{3,11,...}$	$TWIST_{4,12,...}$	$TWIST_{5,13,...}$	$TWIST_{6,14,...}$	$TWIST_{7,15,...}$	$TWIST_{8,16,...}$
1	11	21	31	41	51	61	71 80

LEADING EDGE HEIGHT CARDS

$HL_{1,9,...}$	$HL_{2,10,...}$	$HL_{3,11,...}$	$HL_{4,12,...}$	$HL_{5,13,...}$	$HL_{6,14,...}$	$HL_{7,15,...}$	$HL_{8,16,...}$
1	11	21	31	41	51	61	71 80

JET DEFLECTION CARDS

$DJ_{1,9,...}$	$DJ_{2,10,...}$	$DJ_{3,11,...}$	$DJ_{4,12,...}$	$DJ_{5,13,...}$	$DJ_{6,14,...}$	$DJ_{7,15,...}$	$DJ_{8,16,...}$
1	11	21	31	41	51	61	71 80

CAMBER TYPE FLAG CARD

ICT_1	ICT_2	ICT_3	ICT_4	...	...	...		...	$ICT_{NR\O WS}$	
1					11		21 · 31 · 41 · 51		61	71 · 80

CAMBER ANGLE CARDS

$AC_{1, 9, 17}$	$AC_{2, 10, 18}$	$AC_{3, 11, 19}$	$AC_{4, 12, 20}$	$AC_{5, 13}$	$AC_{6, 14}$	$AC_{7, 15}$	$AC_{8, 16}$
1	11	21	31	41	51	61	71 · 80

(AS MANY AS 10 SETS PERMITTED (1 < NCT < 10))

TRAILING EDGE CAMBER ANGLE CARDS

$ACTE_{1, 9, ...}$	$ACTE_{2, 10, ...}$	$ACTE_{3, 11, ...}$	$ACTE_{4, 12, ...}$	$ACTE_{5, 13, ...}$	$ACTE_{6, 14, ...}$	$ACTE_{7, 15, ...}$	$ACTE_{8, 16, ...}$
1	11	21	31	41	51	61	71 · 80

(NRØWSJ VALUES REQUIRED)

HINGE SECTION TYPE CARD

$IHT_{RØW_1}$	....	$IHT_{RØW_{NRØWS}}$	(HIGHEST VALUE INPUT = NHT)

Columns: 1, 11, 21, 31, 41, 51, 61, 71, 80

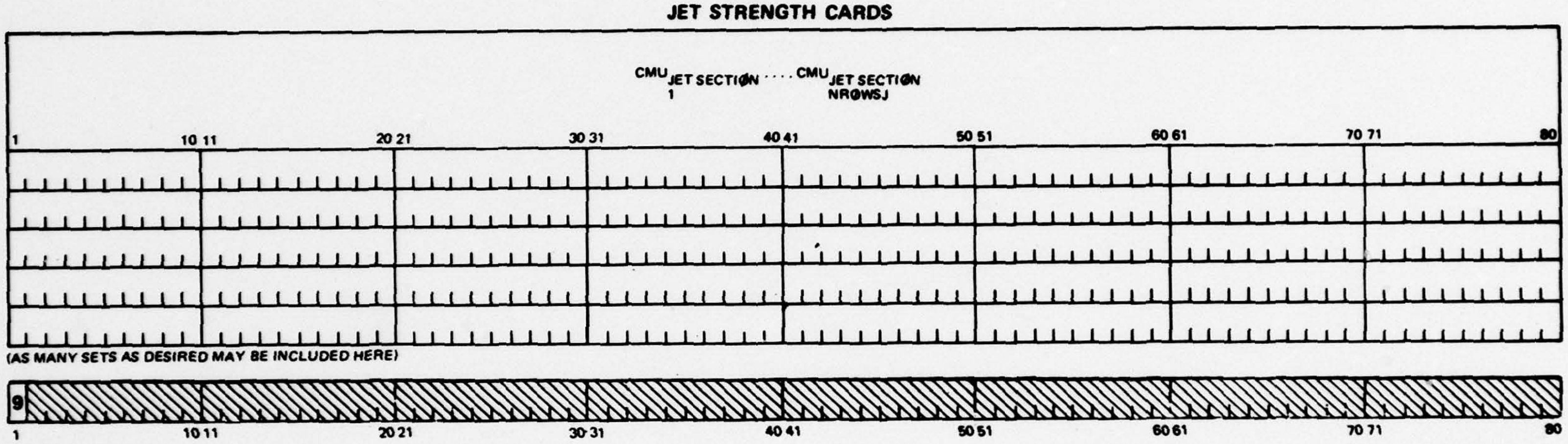
HINGE PARAMETER CARDS

XBH_1	ILT_1	$BETA_1$	XBH_2	ILT_2	$BETA_2$	XBH_3	ILT_3	$BETA_3$	XBH_4	ILT_4	$BETA_4$
1–10	11	12–20	21–30	31	32–40	41–50	51	52–60	61–70	71	72–80

COMPOSITE CASE CARDS

N_1	FACTØR N_1	N_2	FACTØR N_2	N_3	FACTØR N_3	N_4	FACTØR N_4	N_5	FACTØR N_5	N_6	FACTØR N_6	N_7	FACTØR N_7	N_8	FACTØR N_8	N_9	FACTØR N_9	N_{10}	FACTØR N_{10}
1 2	3 8	9 10	11 16	17 18	19 24	25 26	27 32	33 34	35 40	41 42	43 48	49 50	51 56	57 58	59 64	65 66	67 72	73 74	75 80
9																			

JET STRENGTH CARDS



DISTRIBUTION LIST - LIMITED
(VOLUME II ONLY)

Chief of Naval Research
Department of the Navy
Arlington, Virginia 22217
Attn: Aeronautics, Code 461 (TLW) (2 cys)

Naval Air Systems Command
Department of the Navy
Washington, D. C. 20360
Attn: NAVAIR 320 (1 cy)
NAVAIR 5301 (1 cy)

Naval Ship Reasearch & Development Center
Aviation and Surface Effects Department
Bethesda, Maryland 20034
Attn: Director, Aviation Department (2 cys)

Defense Documentation Center
Cameron Station, Building 5
Alexandria, Virginia 22314 (4 cys)

THIS PAGE INTENTIONALLY LEFT BLANK

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)
McDonnell Douglas Corporation
Douglas Aircraft Company
Long Beach, California 90846

2a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

2b. GROUP

3. REPORT TITLE
A THEORETICAL METHOD FOR CALCULATING THE AERODYNAMIC CHARACTERISITICS OF ARBITRARY JET-FLAPPED WINGS. VOL. II EVD JET-WING COMPUTER PROGRAM USER'S MANUAL

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Final Technical Report

5. AUTHOR(S) (First name, middle initial, last name)
Michael L. Lopez, Cheng-Chung Shen, Norman F. Wasson

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
May, 1973	113	2

8a. CONTRACT OR GRANT NO.
N00014-71-C-0250

b. PROJECT NO.

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)
Report No. MDC J5519-02 Vol. II

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT
VOL. II - Approved for public release; distribution unlimited

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY
Office of Naval Research
Aeronautic, Code 461
800 N. Quincy Street
Arlington, Virginia 22218

13. ABSTRACT

This report describes the EVD Jet-Wing Computer Program, which is based upon the Elementary Vortex Distribution (EVD) Jet-Wing Lifting Surface Theory described in Volume I of this report. This program provides a capability for determining the aerodynamic characteristics of wings of arbitrary planform, and includes the following:

1. Spanwise and chordwise loading
2. Spanwise variation of induced drag
3. A capability to investigate the effects of:
   a. Part span flaps
   b. Part span blowing
   c. Pitching, rolling, yawing, and sideslip
4. Total lift and induced drag (momentum method), pitching, yawing and rolling moments, etc.

The program has the capabilities for investigating the effects of a variation of leading and trailing flap deflection, camber, twist, jet deflection, and jet momentum.

DD FORM 1 NOV 65 1473

UNCLASSIFIED
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
AERODYNAMIC STABILITY						
AERODYNAMICS						
BLOWING						
CIRCULATION CONTROL						
COMPUTER PROGRAM						
FINITE ELEMENT METHODS						
FLUID DYNAMICS						
HIGH LIFT SYSTEMS						
JET FLAP						
KUTTA CONDITION						
LIFTING SURFACE THEORY						
NUMERICAL ANALYSIS						
SHORT TAKEOFF AND LANDING AIRCRAFT						
VORTICITY						
WING						